

# Course Outline

- Basic Network Concepts
- Java Overview
- Internet Addressing
- Streams
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- Multi-Threaded Applications
- HTTP



Eng. Asma Abdel Karim  
Computer Engineering Department

7

باللغة العربية مستوى الـ App layer

## Networks and Internet Programming

### Basic Network Concepts



Eng. Asma Abdel Karim  
Computer Engineering Department

1

# Outline

- What is a Network?
- The Layers of a Network.
- IP, TCP and UDP.
- IP Addresses and Domain Names.
- Ports.
- The Internet.
- Firewalls.
- Proxy Servers.



Eng. Asma Abdel Karim  
Computer Engineering Department

2

## What is a Network?

- \*• A **network** is a collection of computers and other devices that can send data to and receive data from one another.
- \*• Connectivity:
  - ✓ – Wires - electromagnetic waves.
  - ✓ – Wireless - radio waves.
  - ✓ – Fiber-optic cables - light waves.
- \*• Such connections carry data between one point in the network and another. This data is represented as bits of information (ON/OFF, 0/1).

البيانات بالنهاية تتكون من 0 و 1



Eng. Asma Abdel Karim  
Computer Engineering Department

3

\* ال Switch يوصل ل data link layer

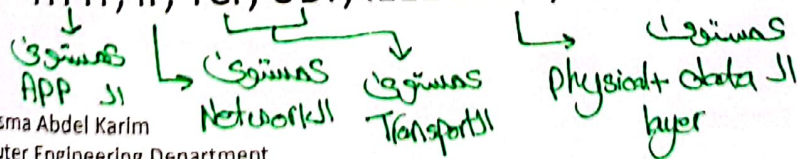
\* ال router يوصل ل Network layer

## What is a Network? (Cont.)

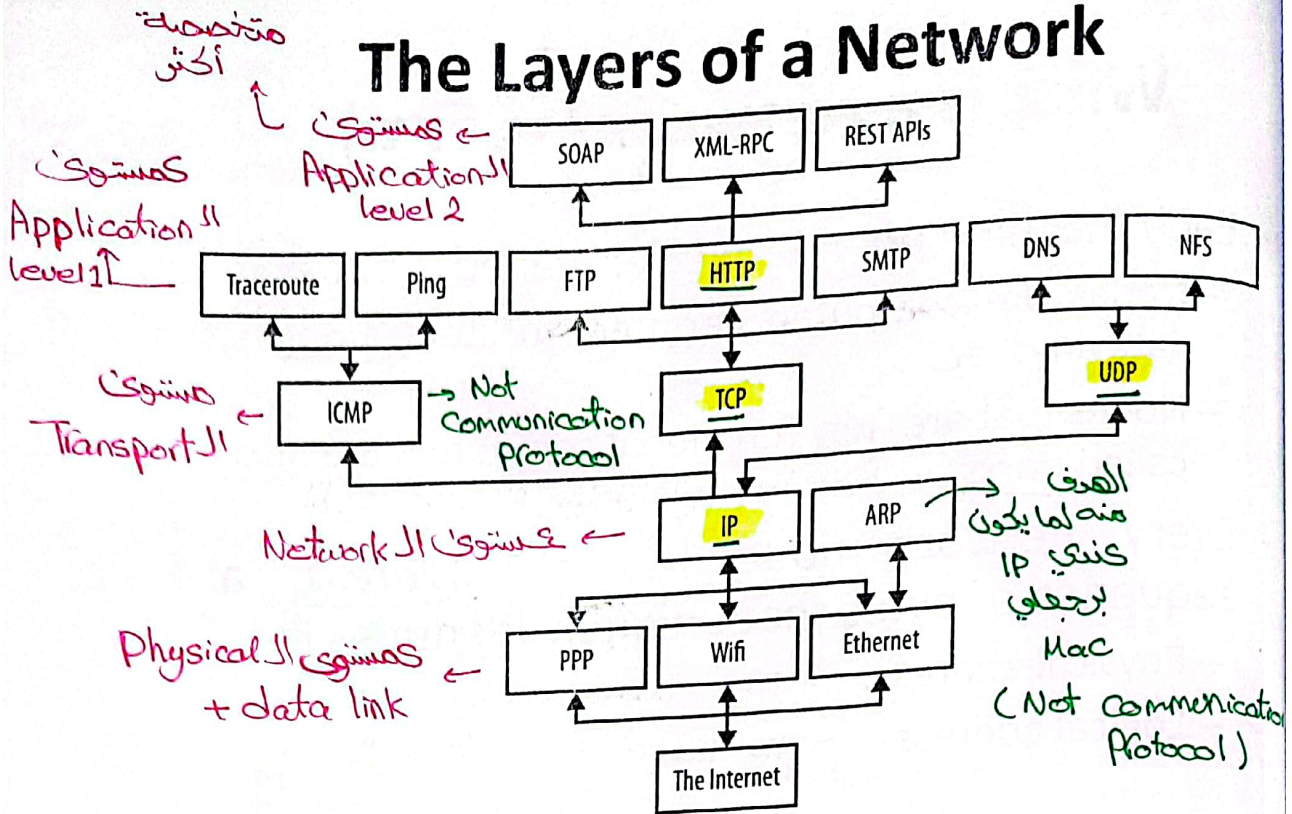
- ✓ Each machine on a network is called a node.
  - Nodes are computers, printers, routers, bridges, gateways, etc...
  - Nodes that are fully functional computers are also called hosts.
- Every network node has an address, a sequence of bytes that uniquely identifies it.
  - ✓ - Physical address. → Mac Address
  - ✓ - Logical address. → IP Address → الذي بنحتاجه

## What is a Network? (Cont.)

- All modern computer networks are packet-switched networks.
  - Data traveling on the network is broken into chunks called packets and each packet is handled separately. *موجودة بكل layer*
  - Each packet contains information about who sent it and where it's going. → Source IP and destination IP
- A protocol is a precise set of rules defining how computers communicate. → كل layer موجود بغير القواعد التي يتحكم *لهذا المستوى هذا ال layer*
  - The format of addresses, how data is split into packets, and so on.
  - HTTP, IP, TCP, UDP, IEEE 802.3, etc...

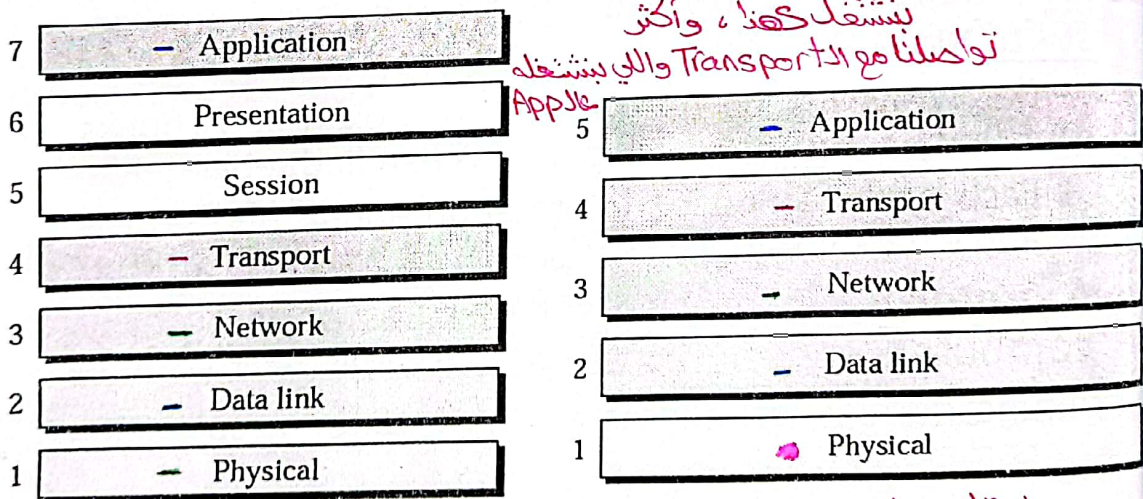


# The Layers of a Network



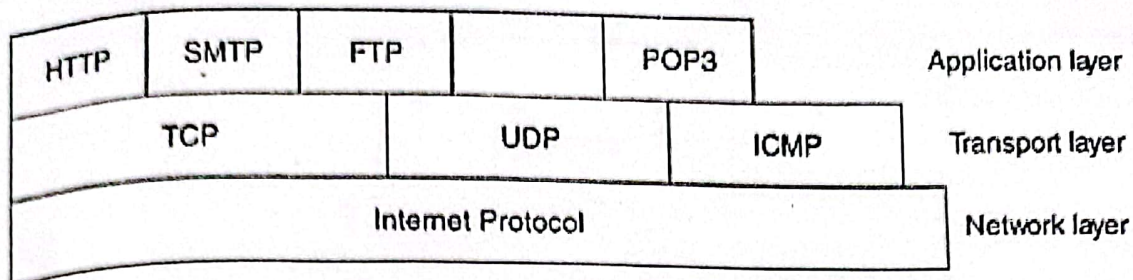
Eng. Asma Abdel Karim  
Computer Engineering Department

# The Layers of a Network OSI Model vs. TCP/IP Model



Eng. Asma Abdel Karim  
Computer Engineering Department

# TCP/IP Protocol Suite Layers



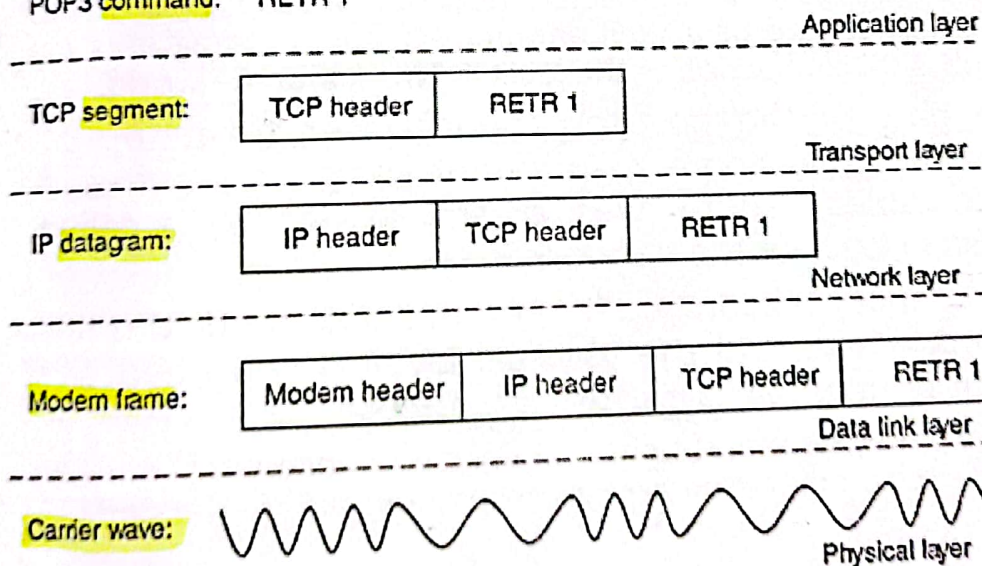
Eng. Asma Abdel Karim  
Computer Engineering Department

8

## TCP/IP Protocol Suite Layers (Example)

*Retrieval command*  
*Email*

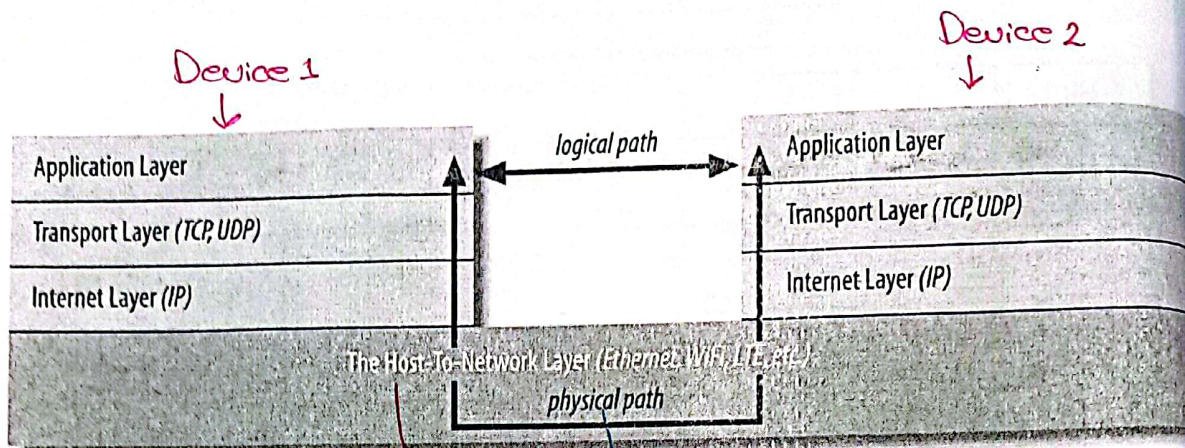
POP3 command: RETR 1



Eng. Asma Abdel Karim  
Computer Engineering Department

9

# The Layers of a Network (Cont.)



يربطها الجوزان تبعاء مع الشبكة.

في كثير من الأحيان بص

فيكون ال Packet ليوصل لل Destination

الذي نلاحظه إنه اللي يطلع من Device 1 هو اللي يوصل ل Device 2 تقريباً بالزبط، كأنه تنوياني إنه في بيديهم logical path . TCP تقريباً



Eng. Asma Abdel Karim  
Computer Engineering Department

بفقد يجعلك هذا الخيال لأنه بيخفي عنك كل العوائق .  
أما ال UDP فلا بيظهر لك العوائق .

10

## The Host-To-Network Layer

- The link layer, data link layer, or network interface layer.
  - Defines how a particular network interface—such as an Ethernet card or a WiFi antenna—sends IP datagrams over its physical connection to the local network and the world.
- The part of the host-to-network layer made up of the hardware that connects different computers (wires, fiber-optic cables, radio waves, or smoke signals) is called the physical layer of the network.
- The primary reason you'll need to think about the host-to-network layer and the physical layer, if you need to think about them at all, is performance.
  - However, whichever physical links you encounter, the APIs you use to communicate across those networks are the same.



Eng. Asma Abdel Karim  
Computer Engineering Department

11

# تكنولوجيا النقل IP The Internet Layer

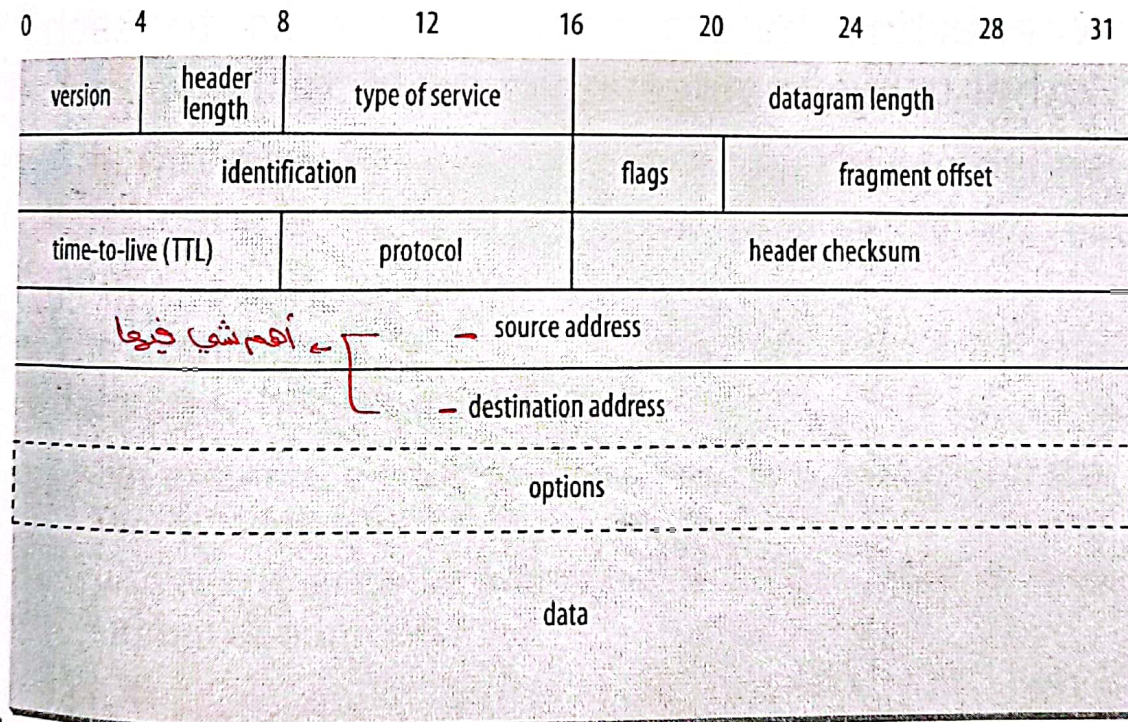
- A network layer protocol defines:
  - how bits and bytes of data are organized into the larger groups called packets, and
  - the addressing scheme by which different machines find one another.
- The **Internet Protocol (IP)** is the most widely used network layer protocol in the world and **the only network layer protocol Java understands**.
- In fact, it's two protocols: تتكون من  
بالأغلب
  - IPv4, which uses 32-bit addresses, and
  - IPv6, which uses 128-bit addresses and adds a few other technical features to assist with routing.
- In both IPv4 and IPv6, data is sent across the internet layer in packets called datagrams. destination Ip و Source Ip في IPv4 / IPv6



Eng. Asma Abdel Karim  
Computer Engineering Department

12

## The Internet Layer IPv4 Datagram Format



Eng. Asma Abdel Karim  
Computer Engineering Department

13

من الواجب Packet كالتالي  
! نفس ال Source App وال  
Destination App ممكن تكون  
بمختلفة Roles

# The Internet Layer

## ↑ Data Transmission Using Packets

- \* • Packets may take different routes to reach the destination depending on the routing approach and congestion level of the network.
- \* • Mechanism to ensure that no packets are lost is available depending on the protocol used to send the data.

ممكن بعض ال Packets تضيع Lost



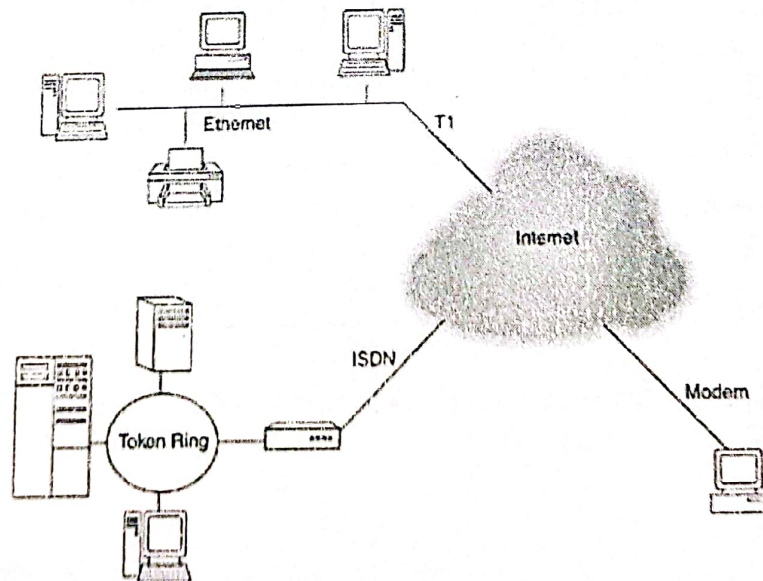
Eng. Asma Abdel Karim  
Computer Engineering Department

1

## The Internet Layer (Cont.)

أجهزة مختلفة يابوس

- The internet layer is responsible for connecting heterogenous networks to each other using homogeneous protocols.



Eng. Asma Abdel Karim  
Computer Engineering Department



# The Transport Layer

وأنا مستوى ال App  
بي أنزل للشبكة

directly ما بي استخدم App protocol فيستخدم TCP أو UDP.

- There are two primary protocols at this level:
- The Transmission Control Protocol (TCP):
  - A reliable protocol.
  - A high-overhead protocol that allows for retransmission of lost or corrupted data and delivery of bytes in the order they were sent. *بجني كل مشاكل الشبكة*
- The User Datagram Protocol (UDP):
  - An unreliable protocol.
  - Does not guarantee that packets are delivered in the correct order (or at all). *→ lower overhead / cost*

بستهلك Cost  
و Delay  
عالية App



Eng. Asma Abdel Karim  
Computer Engineering Department

16

# The Application Layer

هو الذي  
لبيشغل عليه

هو ال interface بين ال user والشبكة.

- ✓ The layer that delivers data to the user.
- ✓ The three lower layers all work together to define how data is transferred from one computer to another.
- ✓ The application layer decides what to do with the data after it's transferred.
  - For example, an application protocol like HTTP (for the World Wide Web) makes sure that your web browser displays a graphic image as a picture, not a long stream of numbers.



Eng. Asma Abdel Karim  
Computer Engineering Department

17

# The Application Layer Protocols

Protocol		Purpose
HTTP	→	Web
SMTP, POP, IMAP	→	Email
FTP, FSP, TFTP	→	File Transfer
NFS	→	File Access
Gnutella, BitTorrent	→	File Sharing
SIP and Skype	→	Voice Communication

↓  
 جميع ال rules لكل Purpose



Eng. Asma Abdel Karim  
 Computer Engineering Department

18

## IP, TCP, and UDP

←  
 شبكة الانترنت  
 Network

- **IP** was designed:
  - To allow multiple routes between any two points and to route packets of data around damaged routers.
  - To be open and platform-independent.
- Packets that make up a particular data stream may not all take the same route.
- Furthermore, they may not arrive in the order they were sent, if they even arrive at all.
- To improve on the basic scheme, **TCP** was layered on top of IP to:
  - Give each end of a connection the ability to acknowledge receipt of IP packets and request retransmission of lost or corrupted packets.
  - Allow the packets to be put back together on the receiving end in the same order they were sent.

← فيو acknowledge

← out of order وصلوا

← ال TCP بتعطيني ترتيب الصحيح



Eng. Asma Abdel Karim  
 Computer Engineering Department

19

موال UDP بتكون سبب المشكلة ، الشبكة هي السبب ولكن ال UDP  
لا تقدم خدمات لتحسين هذه المشكلة مثل ال TCP.

## IP, TCP, and UDP (Cont.)

- TCP, however, carries a fair amount of overhead.
- UDP is an unreliable protocol that does not guarantee that packets will arrive at their destination or that they will arrive in the same order they were sent.
- Although this would be a problem for uses such as file transfer, it is perfectly acceptable for applications where the loss of some data would go unnoticed by the end user.
  - For example, losing a few bits from a video or audio signal won't cause much degradation; it would be a bigger problem if you had to wait for a protocol like TCP to request a retransmission of missing data.
  - Furthermore, error-correcting codes can be built into UDP data streams at the application level to account for missing data.

ما جالدة ضياو هذا ال Frame البسيط

الحالة اللي ممكن نستخدم فيها ال UDP



## IP, TCP, and UDP (Cont.)

Protocol supporting IP

- A number of other protocols can run on top of IP.
- The most commonly requested is ICMP, "the Internet Control Message Protocol," which uses raw IP datagrams to relay error messages between hosts.
  - The best-known use of this protocol is in the ping program.
  - Java does not support ICMP, nor does it allow the sending of raw IP datagrams.
- The only protocols Java supports are TCP and UDP, and application layer protocols built on top of these.
- All other transport layer, internet layer, and lower layer protocols such as ICMP, IGMP, ARP, RARP, RSVP, and others can only be implemented in Java programs by linking to native code.

مش موجودين بال اصغر



\* الذي يعني هو ال Addressing .

## IP Addresses and Domain Names

- As a Java programmer, you don't need to worry about the inner workings of IP, but you do need to know about **addressing**.
- Every computer on an IPv4 network is identified by a unique four-byte address. = 32 bits
  - This is normally written in a *dotted quad format* like 199.1.32.90, where each of the four numbers is one unsigned byte ranging in value from 0 to 255.
- When data is transmitted across the network, the packet's header includes the address of the machine for which the packet is intended (the destination address) and the address of the machine that sent the packet (the source address).



Eng. Asma Abdel Karim  
Computer Engineering Department

22

## IP Addresses and Domain Names (Cont.)

- Routers along the way choose the best route on which to send the packet by inspecting the destination address. The source address is included so the recipient will know who to reply to. *له الذي وصله المسج ممكن يصير به يعمل Reply*
- A slow transition is under way to **IPv6**, which will use 16-byte addresses.
  - This provides enough IP addresses to identify every person, every computer, and indeed every device on the planet.
  - IPv6 addresses are customarily written in eight blocks of four hexadecimal digits separated by colons, such as FEDC:BA98:7654:3210:FEDC:BA98:7654:3210



Eng. Asma Abdel Karim  
Computer Engineering Department

23

# IP Addresses and Domain Names (Cont.)

ترتيب بين ال address IP والاسم الخاص فيه

- Although computers are very comfortable with numbers, human beings aren't very good at remembering them.
- Therefore, the Domain Name System (DNS) was developed to translate hostnames that humans can remember, such as "www.oreilly.com," into numeric Internet addresses such as 208.201.239.101.   
 *لانه يكونوا بتأريين ال IP Address*
- Some computers, especially servers, have fixed addresses.   
 *لانه مقيمة*
- Others, especially clients on local area networks and wireless connections, receive a different address every time they boot up, often provided by a DHCP server.
- \* Mostly you just need to remember that IP addresses may change over time, and not write any code that relies on a system having the same IP address.   
 *الاستخدام الأسماء أفضل من استخدام الأرقام.*

Eng. Asma Abdel Karim  
Computer Engineering Department

# IP Addresses and Domain Names (Cont.)

② حتى لو ما عملت 24 نت بتقدر تفحص ال code تترك

- Some IPv4 addresses can be used on internal networks, but no host using addresses in these blocks is allowed onto the global Internet.   
 *Private / non routable Addresses*
- Addresses that begin with 10., 172.16. through 172.31. and 192.168.   
 *أفحص ال Code بجعل ال Network ال Problems*
- These non-routable addresses are useful for building private networks that can't be seen on the Internet.
- IPv4 addresses beginning with 127 (most commonly 127.0.0.1) always mean the local loopback address.   
 *حتى تمن App ال Network ال Transport وعند ال Network لما يبتوف ال destination 127.0.0.1 يرجع ال packet لنفس ال Source*
- The hostname for this address is often localhost.
- In IPv6, 0:0:0:0:0:0:0:1 (a.k.a. ::1) is the loopback address.
- The IPv4 address that uses the same number for each of the four bytes (i.e., 255.255.255.255), is a broadcast address.   
 *لما يبعث لكل العنود الي موجودة بنفس ال LAN*
- Packets sent to this address are received by all nodes on the local network, though they are not routed beyond the local network.

Eng. Asma Abdel Karim  
Computer Engineering Department

## Ports

← الطريقة التي يعمل فيها Addressing  
مستوى الـ Transport layer.

- Different types of traffic on a computer are sorted out using ports. 16 bit
- Each port is identified by a number between 1 and 65535.
- Port numbers between 1 and 1023 are reserved for well-known services like FTP, HTTP, and IMAP.

\* تستخدم من 1024 لـ 65535 .  
\* Socket معناها بوابة .



Eng. Asma Abdel Karim  
Computer Engineering Department

26

## The Internet

- The world's largest IP-based network.
  - An amorphous group of computers in many different countries on all seven continents (Antarctica included) that talk to one another using IP protocols.
- \* • Each computer on the Internet has at least one IP address by which it can be identified.
  - Many of them also have at least one name that maps to that IP address.

\* DNS : يتعمل لـ Translate الـ Names ليـ بيرواـ Numbers

↓  
استعمال اسم  
من الأرقام .



Eng. Asma Abdel Karim  
Computer Engineering Department

# Internet Address Blocks

كيف الأجهزة  
→ Address

- Blocks of IPv4 addresses are assigned to Internet service providers (ISPs) by their regional Internet registry.
- When a company or an organization wants to set up an IP-based network connected to the Internet, their ISP assigns them a block of addresses.
- Each block has a fixed prefix <sup>①</sup> → Network Address  
ثابت الكد الأجزاء التي على الشبكة  
② الجزء الثاني هو id المميز المنصوص لهذا الجواز جوا الشبكة (host part)
- For instance if the prefix is 216.254.85, then the local network can use addresses from 216.254.85.0 to 216.254.85.255.
- Because this block fixes the first 24 bits, it's called a /24.
- Keep in mind that you have two fewer available addresses than you might first expect:
  - \* ⊖ The lowest address in all block used to identify the network itself, and
  - \* ⊖ The largest address is a broadcast address for the network.



Eng. Asma Abdel Karim  
Computer Engineering Department

28

## Network Address Translation

- In NAT-based networks most nodes only have local, non-routable addresses selected from either : + Private
  - 10.x.x.x,
  - 172.16.x.x to 172.31.x.x, or
  - 192.168.x.x.
- The routers that connect the local networks to the ISP translate these local addresses to a much smaller set of routable addresses.



Eng. Asma Abdel Karim  
Computer Engineering Department

29

عكازة نجران بحد اي Traffic مشهور .

## \* غالبية ال OS بيتجي معالم Firewalls

- The hardware and software that sit between the Internet and the local network, checking all the data that comes in or out to make sure it's safe.
- The firewall can be:
  - part of the router that connects the local network to the broader Internet and may perform other tasks, such as network address translation.
  - a separate machine.
- Modern operating systems like Mac OS X and Red Hat Linux often have built-in personal firewalls that monitor just the traffic sent to that one machine.
- Either way, the firewall is responsible for inspecting each packet that passes into or out of its network interface and accepting it or rejecting it according to a set of rules.



Eng. Asma Abdel Karim  
Computer Engineering Department

30

## Firewalls (Cont.)

- \* • Filtering is usually based on network addresses and ports.
  - All traffic coming from the Class C network 193.28.25.x may be rejected because you had bad experiences with hackers from that network in the past.
  - Outgoing SSH connections may be allowed, but incoming SSH connections may not.

- \* • More intelligent firewalls look at the contents of the packets to determine whether to accept or reject them. → لفريق حسب نوع ال Firewall

← شغل  
Network  
Administer

- The exact configuration of a firewall—which packets of data are and to pass through and which are not—depends on the security needs of an individual site.

لبيخدمين ال data الي مسموح تطلع أو تدخل



Eng. Asma Abdel Karim  
Computer Engineering Department

من الشبكة ويتعمد عال  
Security بنقت الشبكة

31



بالنسبة لنا Network programmers يجب علينا تجنب الـ Firewall  
او يدي اشغل برنامجي  
على الشبكة.

## Firewalls (Cont.)

- The firewall is an excellent tool for network administrators but not for network developers.
- \* – Most corporate firewalls block direct UDP and TCP access.
- \* – Hence, developers must make a choice – either
  - ① use standard Internet protocols and ignore users who work behind firewalls, or adapt software to proxy requests using protocols such a HTTP. ②

Some Entity doing something behind of you



Eng. Asma Abdel Karim  
Computer Engineering Department

خادم يعمل الخدمة بدل عنك

32

## Firewalls (Cont.)

- The firewall is an excellent tool for network administrators but not for network developers.
- Most corporate firewalls block direct UDP and TCP access.
- Hence, developers must make a choice – either use standard Internet protocols and ignore users who work behind firewalls, or adapt software to proxy requests using protocols such a HTTP.



Eng. Asma Abdel Karim  
Computer Engineering Department

33

← اللي بترق الشبكة حيشوفوا الطلب جاي من ال Proxy مش مني

← هو ما بقسملك الخدمة، بطلب الخدمة

مثال ٩.

## Proxy Servers

- If a firewall prevents hosts on a network from making direct connections to the outside world, a proxy server can act as a go-between.
  - Thus, a machine that is prevented from connecting to the external network by a firewall would make a request for a web page from the local proxy server instead of requesting the web page directly from the remote web server.
  - The proxy server would then request the page from the web server and forward the response back to the original requester.
- ① • One of the security advantages of using a proxy server is that external hosts only find out about the proxy server.
  - They do not learn the names and IP addresses of the internal machines, making it more difficult to hack into internal systems.

← لأنه هو  
اللي يكون  
بالوجهة  
فوننا آمن  
أكثر.

② caching → بعمل Save لأي طلب من قبل عثمان  
لو رجع انطلب يكون مخزن



Eng. Asma Abdel Karim  
Computer Engineering Department

34

## Proxy Servers (Cont.)

← عادة بوصولوا بين لمستوى  
ال Transport

- Whereas firewalls generally operate at the level of the transport or internet layer, proxy servers normally operate at the application layer. → أعرف
- A proxy server has a detailed understanding of some application-level protocols, such as HTTP and FTP. → قدرة يفهم المسج اللي جوا ال Packet
- Packets that pass through the proxy server can be examined to ensure that they contain data appropriate for their type.
  - For instance, FTP packets that seem to contain Telnet data can be rejected. ← مش شرط (Telnet) بس



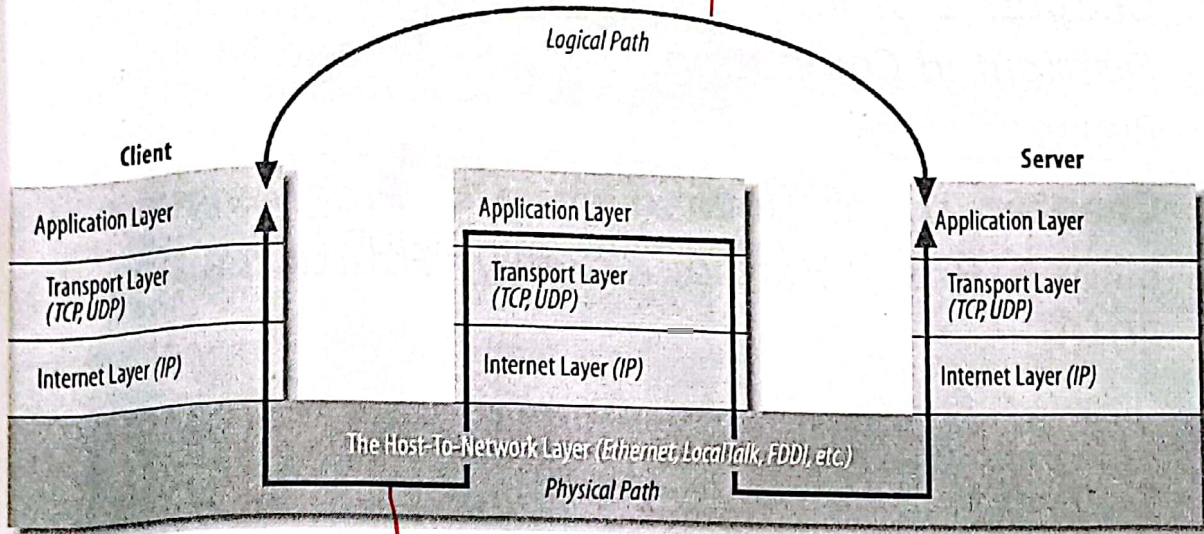
Eng. Asma Abdel Karim  
Computer Engineering Department

35

فيه كذا layers . → بيشغل لحد ال App layer

## Proxy Servers (Cont.)

ما ينقلها مباشرة زي هيك



Eng. Asma Abdel Karim  
Computer Engineering Department

عن طريق  
ال Proxy

36

\* ال Caching مع انه بيفيدنا ك NP بس هو سلاح زو حدين

## Proxy Servers (Cont.)

لانه ما على control لقديم لانم  
بفضل ال Name بال Cach .

- Proxy servers can also be used to implement local caching.
- When a file is requested from a web server, the proxy server first checks to see if the file is in its cache. → افضل Timing
  - If the file is in the cache, the proxy serves the file from the cache rather than from the Internet.
  - If the file is not in the cache, the proxy server retrieves the file, forwards it to the requester, and stores it in the cache for the next time it is requested



Eng. Asma Abdel Karim  
Computer Engineering Department

37

# References

**Chapter 1** of *Java™ Network Programming and Distributed Computing*, David Reilly and Michael Reilly.

**Chapter 1** of *Java Network Programming* Elliotte Rusty Harold, O'Reilly, Fourth Edition 2013.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Networks and Internet Programming

Internet Addressing



Eng. Asma Abdel Karim  
Computer Engineering Department

\* حنكتب برامج يتحركي مع الشبكة \*

## Outline

- Local Area Network Addresses.
- Internet Protocol Addresses.
- The Domain Name System. (DNS)
- Internet Addressing with Java.



Eng. Asma Abdel Karim  
Computer Engineering Department

2

## Local Area Network Addresses

- Devices connected to a LAN have their own unique physical or hardware address.
  - This address is useful only in the context of a LAN.
- Java network programmers do not need to be concerned with the details of how data is routed within a LAN.
  - Java does not provide access to the lower-level data link protocols used by LANs.
  - No matter what type of LAN is used, software can be written for it in Java providing it supports TCP/IP.

\* العناوين  
يكون موجود

هو ال IP Address  
Programmer ك ما يعني  
MAC ال Address

← ما يعني نوع ال LAN ك Programmer كثير



Eng. Asma Abdel Karim  
Computer Engineering Department

3

# Internet Protocol Addresses

- Devices having a direct internet connection are allocated a unique IP address.
  - This address is used by the internet protocol to route IP datagrams to the correct location.
- IP addresses may be allocated:
  - \* Statically: IP address is bounded permanently to certain machine.  
← التي يباينهم الأجهزة التي ما بطلوا
  - \* Dynamically: IP address is leased to a particular machine for a certain period.  
← يتغيروا زي Servers ال  
مثلا الاتوب
  - For example in the case of ISP that offers a pool of modems for dial-up connections.
  - Used when many devices require Internet access for limited periods of time.
  - The Dynamic Host Control Protocol (DHCP) provides addresses on demand from a pool of addresses.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Structure of the IP address

← هو الذي حستخدمه

- Under IPv4, the IP address is a 32-bit number made up of four octets (bytes).
- IP addresses are written in dotted decimal notation (e.g. 127.0.0.1). → loopback address
  - Each byte is an unsigned integer between 0 and 255.
- Each IP address consists of two components:
  - \* left ← The network address: a unique identifier of a specific network. → لميزك Network من الثانية
  - \* Right ← The host address: a unique address of the host in the network it belongs to. ↓

لميزك Host من الثانية داخل نفس ال Network

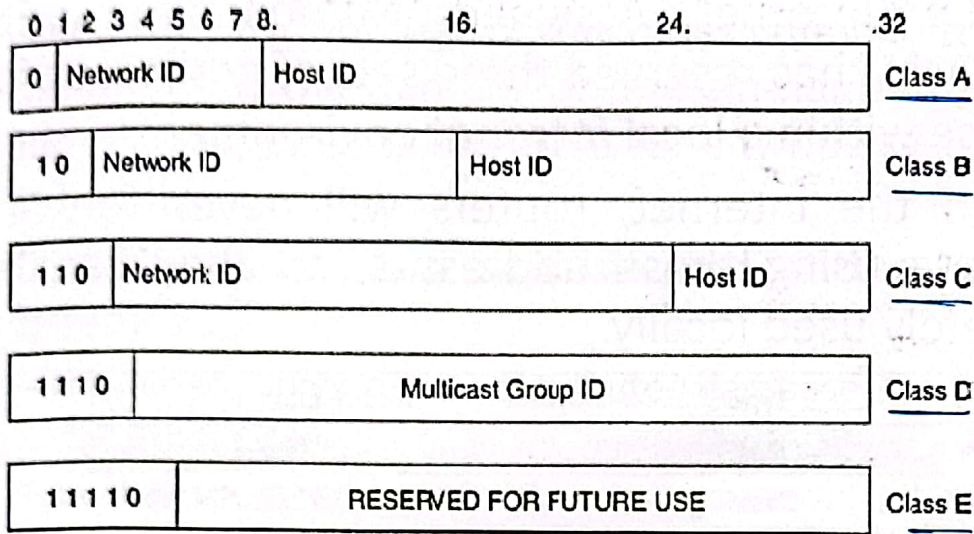


Eng. Asma Abdel Karim  
Computer Engineering Department

- \* all zeros for the Network
- \* all ones for the broadcast

# Structure of the IP address (Cont.)

\* Classful Addresses



لا يستخدم حالياً ، يستخدم Classless



Eng. Asma Abdel Karim  
Computer Engineering Department

6

## Special IP Addresses

تستخدمه كثير بال IP هذا ال Address

- 127.0.0.1 is a special reserved address known as the loopback or localhost address. الاسم المناظر اليه
- The loopback address is very useful when programming and debugging network software.
- Programmers often want to connect to the local machine for testing purposes regardless of whether a connection to the internet exists or not.

\* كل جهاز على الشبكة يمتلك شكل IP Address Name



Eng. Asma Abdel Karim  
Computer Engineering Department

7

## Special IP Addresses (Cont.)

Private Addresses, NAT addresses

- The Internet Assigned Number Authority (IANA) has reserved three sets of addresses for use within a local *intranet* environment.
- On the internet, routers will never forward data using these addresses, so they can be safely used locally.

Type	Address Range
* Class A	10.0.0.0 – 10.255.255.255
* Class B	172.16.0.0 – 172.31.255.255
* Class C	192.168.0.0 – 192.168.255.255



Eng. Asma Abdel Karim  
Computer Engineering Department

أقدر استخدم الأسماء بديل الأرقام.

## The Domain Name System

- Memorizing IP addresses is an impossible task.
- The *Domain Name System (DNS)* makes the internet user-friendly, by associating a textual name with an IP address.
- An entity can apply for a domain name, which can be used by people to locate that entity on the internet.



Eng. Asma Abdel Karim  
Computer Engineering Department



بعضها الاسم وهي بتعطيني ال IP تبعه لتنفيذ الخدمة

## The Domain Name System (Cont.)

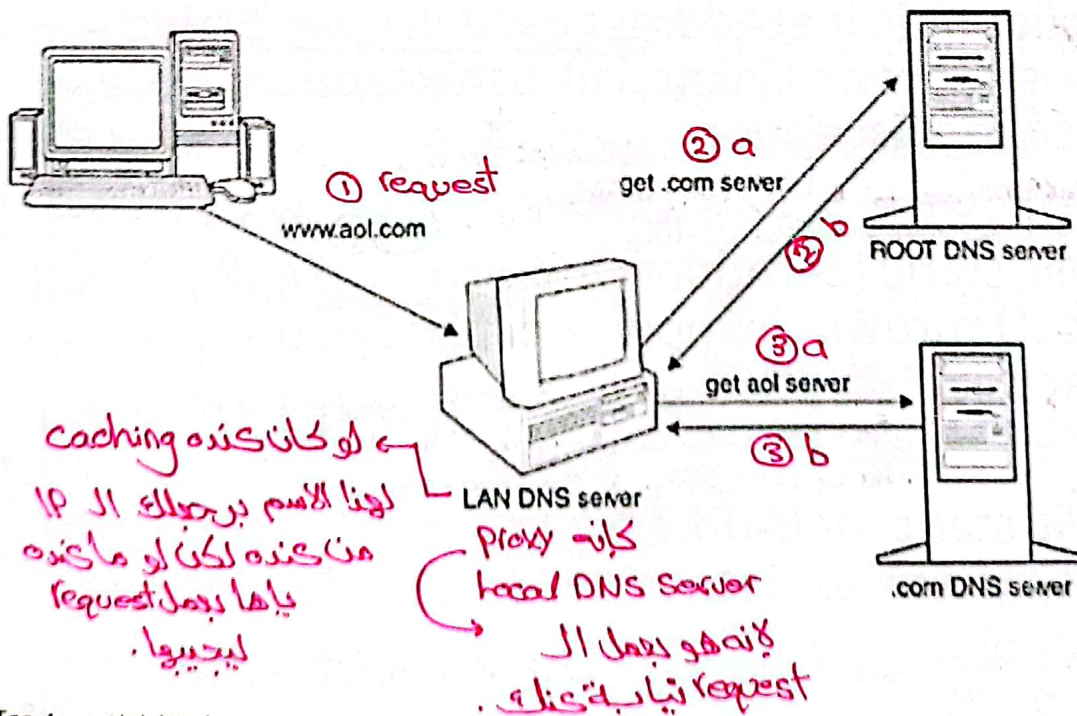
- Given the vast number of machines connected to the internet, the number of domain-name-to-IP-address mappings is too great for any one system to handle.
- The DNS is a more sophisticated and robust system.
  - It can be thought of as a distributed database.
  - Consists of a hierarchical structure which is broken up by the type of address (.net, .com, .gov, .edu, ....) or by the country (.au, .uk, ....).



Eng. Asma Abdel Karim  
Computer Engineering Department

\* مستحيل Server واحد يوسع كل هقول ال mappings .  
 \* بيش أبعت بالهرم تابع ال Servers جزء جزء  
 حسب ال URL اللي ببحث عنه . (من الأنعم الي الأنص)  
 \* فزيك ما بدور بال Servers كاملة وبأخذ وقت زياطة .

## The Domain Name System (Cont.)



Eng. Asma Abdel Karim  
Computer Engineering Department

\* ال Caching مبيح لو بروجي ال IP Address بشكل صحيح  
أما لو بروجي IP غلط فواري مشكلة خاصة انه ما الي عليه Control

## Internet Addressing with Java

- له كيف بنمشل ال IP Addresses بسولة لجهازك أو جواز ثاني بال Java .
- A host on the internet can be represented by either:
    - \* - A dotted decimal format as an IP address, or
    - \* - A hostname such as www.aol.com.
  - Under Java, such addresses are represented by the `java.net.InetAddress` class.   
 IP و Name له ممكن
  - = There are no public constructors for this class.   
 Arbitrary addresses may not be created.
  - = Instead, there are static methods that return `InetAddress` instances.

Class لمثل ال  
Objects ال  
المثل ال  
IP Address تبعه

ملكيته ال Class ككل بقدر انا ديجا باسم

هاي الطريقة  
flexibility بتعطيني



Eng. Asma Abdel Karim  
Computer Engineering Department

actual type تبعه

\* ال InetAddress سب اعمل ال object لحاله  
لبري أي نوع بنرج تحته هذا ال object (IPv4, IPv6) وبخيه ال

## Methods to Create InetAddress

Objects (1)   
 \* بطولوا محل ال Constructors ، بنستعملهم لنعمل objects of type InetAddress   
 Static Method  
 بنوع IP واحد

public static InetAddress getByName (String host) throws UnknownHostException, SecurityException   
 اسم الجواز  
ال ال بي احبيب  
ال IP تبعه

Checked ، بنوعى لو  
DNS عليه  
UnChecked ، بنوعى لو الجواز ثاني  
فشلات وما  
ومثلت بالوقت  
موسموره ينفذ ال operation مثال  
ال DNS و بروجي ال IP وال Name  
ال IP تبعه  
ال DNS و بروجي ال IP وال Name  
ال IP تبعه

public static InetAddress[] getAllByName (String host) throws UnknownHostException, SecurityException   
 ما بروجي بس IP وحدة random ، مفيدة لما اكون عارفة انه هذا الاسم  
بنرجع كل ال IP الي Mapped لنفس هذا الاسم ، مفيدة لما اكون عارفة انه هذا الاسم  
Mapped لاكثر من IP . بنرجعهم ك Array

public static InetAddress getLocalHost () throws UnknownHostException, SecurityException

بنروجي ال InetAddress object الي بتمثل جوازك (زيها زي الباقي بتعمل DNS request)



Eng. Asma Abdel Karim  
Computer Engineering Department

EX % InetAddress address = InetAddress.getByName ("www.aol.com")

له صار يا شعلك www.aol.com

ما بتعمل new زي اول

هذه الـ Methods ما بيعملوا DNS lookup

# Methods to Create InetAddress Objects (2)

Java 1.4 adds two more factory methods that do not check their addresses with the local DNS server. \*

- The first creates an InetAddress object with an IP address and no hostname.

IP V4 يكون حجمه 4 → one parameter ← الترتيب من الشمال لليمين  
 و IP V6 يكون 16  
 دائما Signed  
 بي احنا بنسا الـ address يكون unsigned ← باللاب بعملة

```
public static InetAddress getByAddress (byte[ ] address)
throws UnknownHostException
```

Array خلاصه 4 او 6  
 لـ بتعريفه لو عيطتيا حجم

بدون اسم  
 بتكون

- The second creates an InetAddress object with an IP address and a hostname.

يعطيه اسم و الـ address

```
public static InetAddress getByAddress (String hostName,
byte[] address) throws UnknownHostException
```



لا data fields اللي عندي ← Name  
 يكون عندي  
 . getters

## Getter Methods

- The following method returns the hostname of the InetAddress object:

```
public String getHostName () throws SecurityManagerException
```

هنا السؤال

- The following method returns the IP address of the InetAddress object in byte format. The bytes are returned in network byte order, with the highest byte as byteArray[0]:

```
public byte [] getAddress ()
```

بتعيب الـ address ك Array of bytes من شمال لليمين  
 بي اعلمه check .  
 يستخدمه لما

- The following method returns the IP address of the InetAddress in dotted decimal format:

```
public String getHostAddress ()
```

بتعيب الـ address ك String يستخدمها لما بي اطلع الـ address بشكل مرتب  
 او بي اخطه بـ file .



# Object Class Inherited Methods

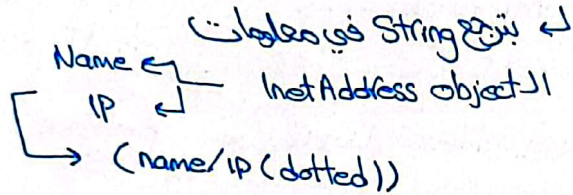
تتقارن بين ال IP Addresses ← بتقارن 2 objects لو متساويين بترجع true لو لا متساويين بترجع false

• public boolean equals (Object o)

• public int hashCode () →

حتمسب ال hash code بناء على IP Address

• public String toString ()



Eng. Asma Abdel Karim  
Computer Engineering Department

## Example 1: Using InetAddress to Determine Localhost Address

← برجلي ال address الذي يمثل جوازبي

```
import java.net.*;

public class LocalHostDemo {

    public static void main(String[] args) {
        System.out.println("Looking up local host!");
        try {
            InetAddress localAddress = InetAddress.getLocalHost();
            System.out.println("IP address: "+localAddress.getHostAddress());
        }
        catch (UnknownHostException uhe){
            System.out.println("Error - unable to resolve localhost");
        }
    }
}
```



Eng. Asma Abdel Karim  
Computer Engineering Department

بدي اجيب ال Inet Address تبع جهاز  
ثاني

## Example2: Using InetAddress to Find Out About Other Addresses

```
import java.net.*;
```

```
public class NetworkResolverDemo {
```

```
public static void main(String[] args) {
```

```
if (args.length==1){ بديها بالزبط ا الي هو الجهاز →
```

```
System.err.println("Syntax - NetworkResolverDemo host");
```

```
System.exit(0);
```

```
}
```

```
System.out.println("Resolving "+args[0]);
```

```
try{
```

```
InetAddress addr = InetAddress.getByNames(args[0]);
```

```
System.out.println(addr);
```

```
}
```

```
catch (UnknownHostException uhe){
```

```
System.out.println("Error - unable to resolve host name");
```

```
}
```

```
}
```

```
}
```

*ال Name الي دخله  
ال user*

*لو استخدمت  
getByName*

*لح ترجملي كل ال IP's  
الي لونها ال Name*

*وستان اظهم لازم اعمل  
for loop*

*for (InetAddress a:addr)  
System.out.println(a);*

*\* ما بجملة run بيون ما ارنزل ال Name الي بديها  
مثل Yahoo.com مثلا.*



Eng. Asma Abdel Karim  
Computer Engineering Department

18

## Address Types Methods

- public boolean *isAnyLocalAddress()*
- public boolean *isLoopbackAddress()*
- public boolean *isLinkLocalAddress()*
- public boolean *isSiteLocalAddress()*
- public boolean *isMulticastAddress()*
- public boolean *isMCGlobal()*
- public boolean *isMCNodeLocal()*
- public boolean *isMCLinkLocal()*
- public boolean *isMCSiteLocal()*
- public boolean *isMCOrgLocal()*

*لو ال address عبارتي  
loopback بترجع T او لا بترجع F  
لو ال address كان NATed  
non routable بترجع T او لا بترجع F  
↓  
Private*

*ل IPv6  
ما بجملة*



Eng. Asma Abdel Karim  
Computer Engineering Department

19

← الهدف منو تعمل انشي بيشبه ال Ping

## ← **Testing Reachability** ← مشاكل ال Reachability كمان هياك ما بنوينا كثير

- public boolean *isReachable* (int timeout) throws IOException → لوعلية الإرسال والإستقبال فشلت بويه
- public boolean *isReachable* (NetworkInterface interface, int ttl, int timeout) throws IOException

↓  
عدد ال hops اللي بينك  
عليك ال Packet

↓  
بال ms



Eng. Asma Abdel Karim  
Computer Engineering Department

\*مش انت تختار ال Version ال ال class من احواله بجددك ياها بس في *instanceof* بتبينك شونوع ال IP تاعك

## **Inet4Address and Inet6Address**

"عائلتين فرعيين من ال *InetAddress*"

- Public final class *Inet4Address* extends *InetAddress*

Ex: `if (addr instanceof Inet4Address)` → نتج true لو  
False او لا

- Public final class *Inet6Address* extends *InetAddress*

\*instance of : بتجول لو كانا نوعه هيك  
↓  
كلمة ورجعة true / false



Eng. Asma Abdel Karim  
Computer Engineering Department

# The NetworkInterface Class

- The NetworkInterface class represents a local IP address. This can be:
  - \* - A physical interface such as an additional Ethernet card, or wifi card.
  - \* - A virtual interface bound to the same physical hardware.
- The NetworkInterface class provides methods to enumerate all the local addresses, regardless of interface, and to create InetAddress objects from them.

بنقدرون طريقها  
نعمل ال  
NetworkInterface  
objects  
ويعين ناخذ  
المصفات تاكينوم

Class بال datafield



Eng. Asma Abdel Karim  
Computer Engineering Department

## Methods to Create NetworkInterface Objects

بمجموعتي Name و InetAddress  
Object جوه فيه IP مخزن ك String

- By name: NetworkInterface getByName (String name) throws SocketException

فالتا يستخدم  
لا Physical address (interface)

بذل من ال layer APP لل layer Data link ستوفالو فعلا

- By IP address: NetworkInterface getByInetAddress (InetAddress address) throws SocketException

هذا الاسم موجود عندها الجواز لو اه برجع Object بعينه، لو لا برجع ال null وانا ماكان  
قادر اصلا يجب المعلومة برمي Exception.

بشكل ال List او array list  
بمقدر اخط فيه Object  
واستخدم عليهم  
Methods تطلوني بيانات  
ال objects

- By enumeration: NetworkInterface getNetworkInterfaces() throws SocketException

لنا بي اعرفي كل ال Network address  
الي موجودة على الجواز و تطلعلي عليهم



Eng. Asma Abdel Karim  
Computer Engineering Department

\* أي object لا يطبقه الشاشة direct بينوع String to التي هم  
 Name / IP واما في Name يكون فاضي أو محطوب في String  
 Method  
 IP //

## Getter Methods

- The following method returns a `java.util.Enumeration` containing an `InetAddress` object for each IP address the interface is bound to:

```
public Enumeration getInetAddresses()
```

- The following method returns the name of a particular `NetworkInterface` object, such as `eth0` or `lo`:

```
public String getName()
```

- The following method returns a more human-friendly name for the particular `NetworkInterface` — something like "Ethernet Card 0.":

```
public String getDisplayName()
```



Eng. Asma Abdel Karim  
 Computer Engineering Department

↓  
 نتجول في اسم  
 كامل Standard  
 معرف من الجافا لو علنا من عايزة مختلفة بخدري  
 ما هو ما بتغير حسب ال OS زي ال getName .

## References

Chapter 3 of *Java™ Network Programming and Distributed Computing*, David Reilly and Michael Reilly.

Chapter 4 of *Java Network Programming*, Elliott Rusty Harold, O'Reilly, Fourth Edition, 2013.



Eng. Asma Abdel Karim  
 Computer Engineering Department



# Networks and Internet Programming

## Data Streams Part-1



Eng. Asma Abdel Karim  
Computer Engineering Department

1

## Outline

- Overview.
- How Streams Work? *نحن تعلم كيف نقرأ ونكتب data من بيت البرنامج ،*
- Low-level Streams. *ما رح نكتب بالشبكة لسا.*
- Filter Streams.



Eng. Asma Abdel Karim  
Computer Engineering Department

2

# Overview

← البرنامج لو بحاجة يكتب أو يقرأ data من برنامجنا محتاج الStreams.

- Communication over networks, with files, and even between applications, is represented in Java by Streams.
- Stream-based communication is central to almost any type of Java application.
- Almost all network communication (except UDP communication) is conducted over streams.

\* ال TCP لما تفتح Socket بيوطيك Streams تكتب وتقرأ عليهم.



Eng. Asma Abdel Karim  
Computer Engineering Department

\* ال UDP لما تفتح Socket ما بيحي Streams ، يعني

ما بيستخدمه بشكل أساسي للكتابة والقراءة بس ممكن تستخدمه قريًا.

## What Exactly are Streams?

- Byte-level communication is represented in Java by data streams.
- Data streams are conduits through which information is sent and received.

يرسل ويستقبل من خلالها  
Virtual قنوات أو أنابيب



هذا ال Stream ، بتيلش ال Data تتفل من ال App ال Destination أو العكس وفيها نوعين

low-level streams → أول خطوة وما بيستفاد منها ، لينتقل معها ب byte level ، ما بتكون مفلتق و لا خالية



Eng. Asma Abdel Karim  
Computer Engineering Department

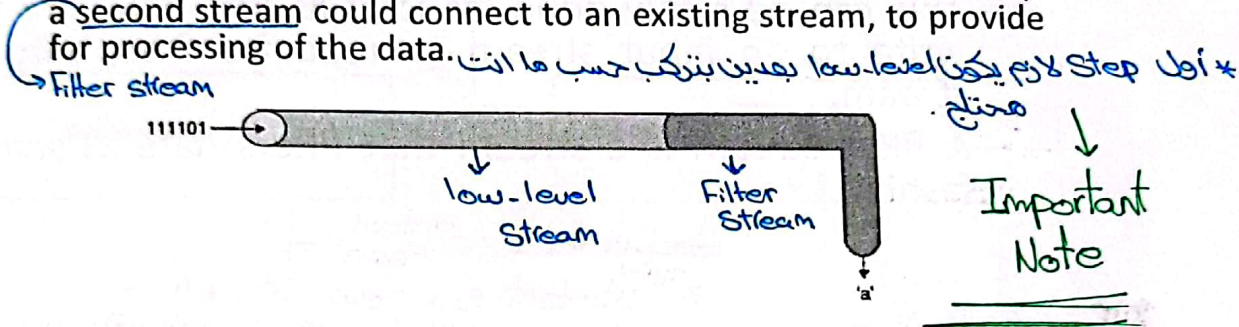
Filter Streams → عشان تسوق البرمجين ، بتغلش ال Data بطريقة معينة

من الشوائب اللي بتخرب ال Data .

معناه العرفي جدول المياه

# What Exactly are Streams? (Cont.)

- When designing a system, the correct stream must be selected.
  - \* - The type of stream used is not important, as a consistent interface is provided.
    - المهم
    - أنه مختلف طريقة
    - الترجع الـ Data بطريقة
    - حيكون في أكثر من طريقة تحل فيها المشكلة
- Streams may be chained together, to provide an easier and more manageable interface.
  - \* - If for example, data needed to be processed in a particular way, a second stream could connect to an existing stream, to provide for processing of the data.
    - صحيحة، أما
    - لحقيقة الحد
    - ما يتفرقا



# What Exactly are Streams? (Cont.)

- Streams are divided into two categories:
  - ✓ - Input streams that may be read from. → أنا بقرا
  - ✓ - Output streams that may be written to. → أنا بكتب
- Although streams are usually one-way, multiple streams can be used together for two-way communication.

ما في stream one بملك two-way communication  
 لو بديك تقرا وتكتب بفتح 2 Streams واحد يقرا  
 وواحد يكتب بنفس الوقت

# What Exactly are Streams? (Cont.)

- في صيغة عالية بال Streams وكلمة بنفس الحجم وكلمة بتركيبها بعض شئ ما كان نفعهم،
- In Java, streams take a flexible, one-size-fits-all approach.
    - \*- They are fairly interchangeable, and can be applied on top of another stream, or even several other streams.
  - You can attach any filter stream to any low level stream (i.e. file or network stream).
    - \*- This can be safely done, as long as you don't try to write to an input stream or read from an output stream.
    - \*- A filter stream is a stream that filters data in some fashion.

• سبب انت  
كلمة تعرف  
تركيب بطريقة  
مع ليتم  
التواصل  
بطريقة  
صح.

declare type ← actual type ←  
\* Student s = new Student  
[ يمكن اي نوع فوهي من ال declare عادي  
\* لو كان Student معرفك abstract ما يقدر اكتب هاي الجملة (new Student)  
سبب declare type عادي يقدر اكتبه.



Eng. Asma Abdel Karim  
Computer Engineering Department

(new Student)

## How Streams Work?

- Streams for reading inherit from a common superclass, the java.io.InputStream class.
- Streams for writing inherit from a common superclass, the java.io.OutputStream class.
- These are abstract classes; they cannot be instantiated.
  - \* - Instead, an appropriate subclass for the task in hand is instantiated.

Subclass بتخير بجزء ال abstract تختار ال



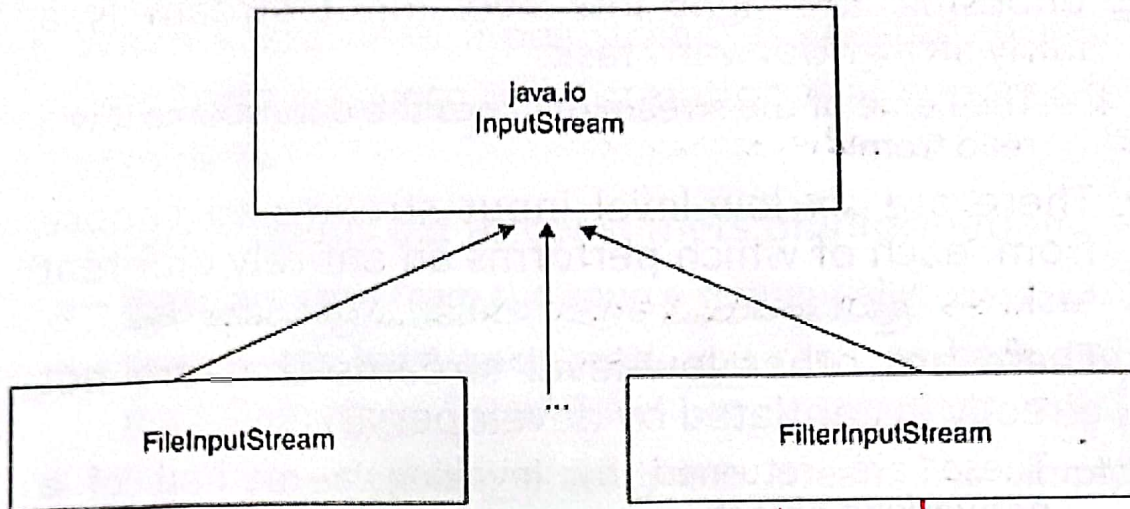
Eng. Asma Abdel Karim  
Computer Engineering Department

اللي بناسب حالة ال  
Object لأنه

ما يقدر يكون من نوع OutputStream أو

InputStream

## How Streams Work? (Cont.)



↳ Subclasses ↵

التي يمكن تخيلها  
التي هي نوع  
التي هي object

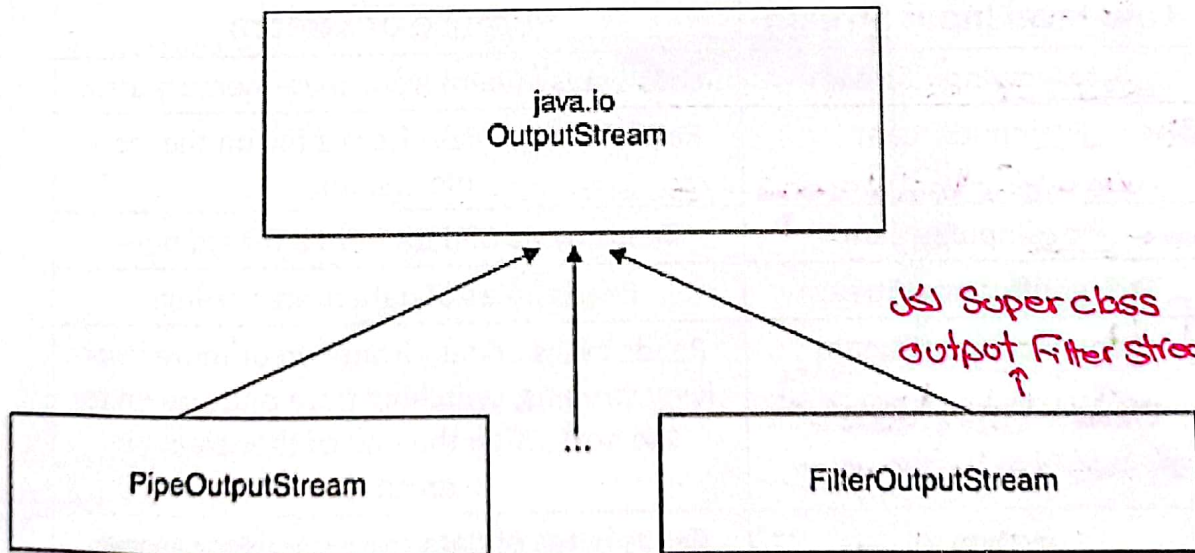
Super class

input Filter stream



Eng. Asma Abdel Karim  
Computer Engineering Department

## How Streams Work? (Cont.)



Super class  
output Filter stream

↳ Subclasses ↵



Eng. Asma Abdel Karim  
Computer Engineering Department

# Reading from an Input Stream

← دائما أول خطوة لتبني قبة يكون level - سطا .

• Choosing the right low-level input stream is a fairly straightforward task.

\* - The name of the stream matches the data source it will read from.   
 \* نحن نعمل العنصر وبنكمله

• There are six low-level input streams to choose from, each of which performs an entirely different task.   
 ← بتحدد النوع حسب ال source التي تتقار منه   
 نوع يعتمد على نوع استخدامك للقراءة فسهل بتحدد النوع

• There are other low-level streams that are not directly instantiated by developers

\* - These are returned by invoking a method of a networking object.

ينبغي وقت ال TCP socket ، جواها ينبغي معها "output/input Streams" مشانت بتعلم



Eng. Asma Abdel Karim  
Computer Engineering Department

11

# Reading from an Input Streams

(Cont.)

كلهم read bytes لا رقم   
 level - سطا ، يفرق ال source التي يتقار منه

يستخدمه بال   
 UDP سبب اختيار ترك   
 FilterStream   
 يحتاج هذا ال Stream   
 بتنكيه ال array

of bytes   
 جوا برنامك   
 (local) ↑

File   
 بتقار من ال local   
 نستخدمه لما ال thread يتكلم مع بعض .

PipedInputStream   
 بتقار من ال local   
 بتقار من ال local

StringBufferInputStream   
 بتقار من ال local   
 بتقار من ال local

SequenceInputStream   
 بتقار من ال local   
 بتقار من ال local

System.in   
 بتقار من ال local   
 بتقار من ال local

Low-level Input Stream	Purpose of Stream
<u>ByteArrayInputStream</u>	Reads bytes of data from an in-memory array.
<u>FileInputStream</u>	Reads bytes of data from a file on the local file system.
<u>PipedInputStream</u>	Reads bytes of data from a thread pipe.
<u>StringBufferInputStream</u>	Reads bytes of data from a string.
<u>SequenceInputStream</u>	Reads bytes of data from two or more low-level streams, switching from one stream to the next when the end of the stream is reached.
<u>System.in</u>	Reads bytes of data from the user console.

تقار data من ال Console   
 معرفها وجاها



Eng. Asma Abdel Karim  
Computer Engineering Department

12

بال   
 Class من Java library   
 سبب ال Instance

البيانات تأتي بال Data مثل ال Files

## Reading from an Input Stream (Cont.)

بقرين طريقا ال Stream أبدا بقراءة ال Data من ال Information Source

• When a low-level input stream is created, it will read from a source of information that supplies it with data.

• Inputs streams act as consumers of information.

\* - Bytes are read from the source sequentially. → دائما بالتسلسل

\* - Once bytes have been read, you can't go back and read them again. بالوضع الطبيعي ال Data ما اليا مكان بتفقد فيه أنته النقل ال

\* - Bytes haven't been erased, the stream has simply moved on to the next byte of information. لو أنت خذتها ب Array

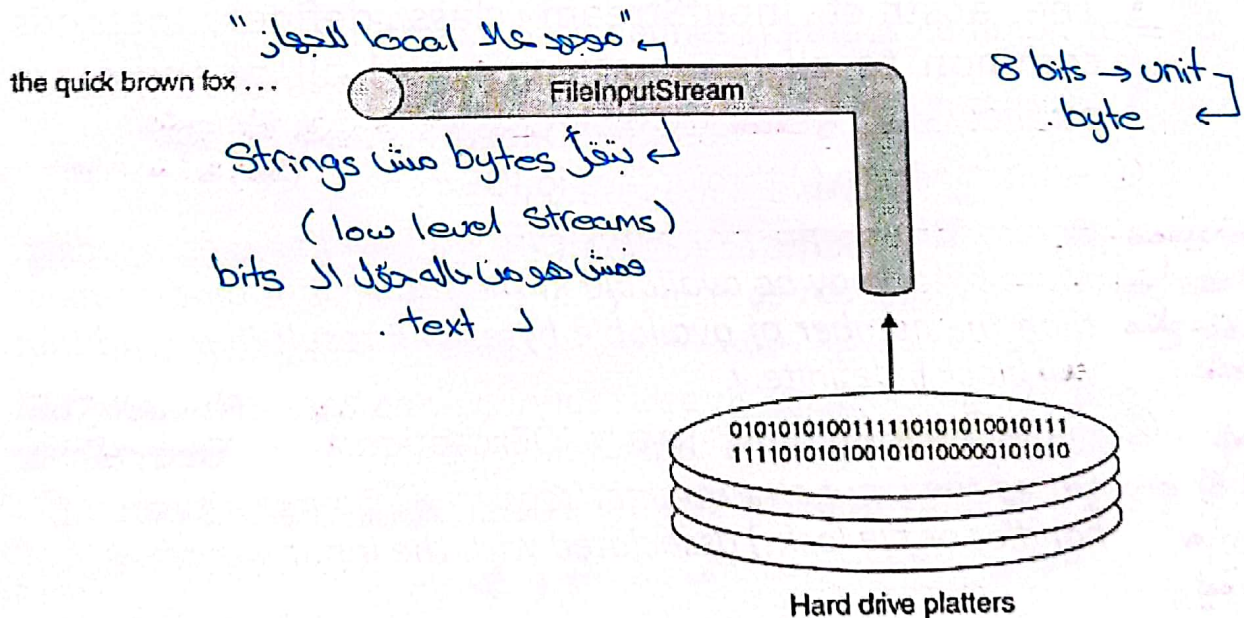


Eng. Asma Abdel Karim  
Computer Engineering Department

ما انجوا ال Bytes بس ما بقدر ارجعهم مرة ثانية لو انقروا بنفس ال Stream ( هذا ال default .

\* بعض الأحيان يكون في buffer ال Stream بجزن فلو كان بي ارجع يرجع ال Data بقدر لانه في buffer ليعنا الخطة لترتيب Stream جديد.

## Reading from an Input Stream (Cont.)



Eng. Asma Abdel Karim  
Computer Engineering Department

# Reading from an Input Stream (Cont.)

• Input streams use blocking I/O.   
 ← سمينتهيك  
 ← لأنه البرنامج  
 ← يتوقف بعد  
 ما يتم عملية  
 ال I/O

• **Blocking I/O** is a term applied to any form of input or output that does not immediately return from an operation.

• Blocking I/O may cause performance problems.   
 ← عملية القراءة يتخاف منها أكثر من الكتابة خاصة لو كانت بين الشبكة لأنه يعتمد على ال Data

\* – This can be alleviated by using **data buffering**.

← or Multithreading  
 ← يقلل من ال blocking



Eng. Asma Abdel Karim  
 Computer Engineering Department

لو كان في Data واحدة أما لو ما في Data يكون فاضي لسا.

15

← ال low level بس هدف بيوزنر أما ال Filter

← جينضاف عليهم  
 Methods  
 زيادة

## The java.io.InputStream Class

• The abstract InputStream class defines methods common to all input streams and all of them are public:   
 ← يترجع int قدييه في bytes لسا ما انقرت بال Stream و

① – **int available ( )** throws java.io.IOException   
 ← جبهة المقراءة (هلا)

← مفيدة عشان  
 ← اقرأ وقد الي  
 ← متاح حتى ما يصير  
 . block

Returns the number of bytes currently available for reading. More bytes may be available in the future, but reading more than the number of available bytes will result in a read that will block indefinitely.

② – **void close ( )** throws java.io.IOException   
 ← Stream ال  
 ← إذا تسكر  
 ← ما يرجع  
 ← ليستخدم  
 ← حتى لو كنتك ال .ref

Closes the input stream and frees any resources (such as file handles or file locks) associated with the input stream.   
 ← لتسكر ال Stream  
 ← وتخليه Free



Eng. Asma Abdel Karim  
 Computer Engineering Department

16



# The java.io.InputStream Class (Cont.)

- by default Stream دائما بتكون مصبوطة أول ال Stream دائما
- ☞ The abstract InputStream class defines methods common to all input streams and all of them are public: → كود ال bytes التي عسوجلي أقرها بعد ما أخط ال mark بتنفذ ال mark تضيغ  
تشفق فقط ال Stream التي فيها buffering →
- ⑤ – void mark (int readLimit) → بس أنا ديها بترجع ال Pointer للمكان الذي طهر ال  
1. Records the current position in the input stream, to allow an input stream to revisit the same sequence of bytes at a later point in the future, by invoking the **InputStream.reset()** method. →
- 2. Not every input stream will support this functionality. **Mark** وبكمل قراءة من هناك
- ④ – boolean **markSupported()** → بتجع لو ال Mark وال reset مدعومة من قبل ال Stream  
1. Returns "true" if an input stream supports the mark() and reset() methods, "false" if it does not.
- 2. Unless over ridden by a subclass of InputStream, the default value returned is false. ↓



Eng. Asma Abdel Karim  
Computer Engineering Department

ال default تكون faulke

17

\* ال `ByteArrayInputStream` مع انه level سا ولكن مشان بقول ال `Data Mem` اني اولاً فل ال Array هاي كاترا ال buffer فويك بتكون بتتم ال `Mark / reset`.

# The java.io.InputStream Class (Cont.)

☞ The abstract InputStream class defines methods common to all input streams and all of them are public:

- ⑤ – **int read()** throws java.io.IOException  
بتجعوا جا `int` من byte
- ① **Returns the next byte of data from the stream.**  
4 bytes ↓
- ② Subclasses of **InputStream** usually override this method to provide custom functionality (such as reading from a file or a string). → كل Method بتعملها override بطريقة المناسبة
- ③ As mentioned earlier, input streams use blocking I/O, and will block indefinitely if no further bytes are yet available.
- ④ When the end of the stream is reached, a value of **-1** is returned.  
← = `FFFFH` كل قيم ال `int` بتكون -1

\* بتجع `int` مشان `byte` مشان ما يفكر مثلا اني هاي القيمة "`FFFF`" هي end of stream لانها مش هيك فما بدين خن نقطة بين ال data الفعلية وبين ال `end of stream`.



Eng. Asma Abdel Karim  
Computer Engineering Department

18

## The java.io.InputStream Class (Cont.)

- The abstract InputStream class defines methods common to all input streams and all of them are public:

⑥ - `int read(byte[] byteArray)` throws `java.io.IOException`

ما بتزجلك  
byte  
وجهة  
بتنوع عدد ال bytes  
اللي انقروا بنجاح ،  
لو وهو بتقرأ فوصلت  
لده end بتزج -1

- Reads a sequence of bytes and places them in the specified byte array, by calling the `read()` method repeatedly until the array is filled or no more data can be obtained.
- This method returns the number of bytes successfully read, or `-1` if the end of the stream has been reached.

\* ملخها blocking فيا ! اما بتقرأ كل ال Array وتتمشى

او إذا فيه byteArray أقدم من حجم ال Array بتقرأ  
اللي يساوي عددها وتتمشى ويكون في جزء  
مؤقت.



Eng. Asma Abdel Karim  
Computer Engineering Department

19

## The java.io.InputStream Class (Cont.)

- The abstract InputStream class defines methods common to all input streams and all of them are public:

⑦ - `int read(byte[] byteArray, int offset, int length)` throws `java.io.IOException`, `java.lang.IndexOutOfBoundsException`

اشه ما  
تطلعها بل حدود  
ال Array

- Reads a sequence of bytes, placing them in the specified array.
- Unlike the previous method, `read(byte[] byteArray)`, this method begins stuffing bytes into the array at the specified offset, and for the specified length, if possible. This allows developers to fill up only part of an array.
- Developers should be mindful that at runtime, out-of-bounds exceptions may be thrown if the array size, offset, and length exceed array capacity.

من طرفنا تبتلي تقبي  
جزء ال Array



Eng. Asma Abdel Karim  
Computer Engineering Department

20

## The java.io.InputStream Class (Cont.)

- The abstract InputStream class defines methods common to all input streams and all of them are public:
- ⑧ – void reset() throws java.io.IOException
    - Moves the position of the input stream back to a preset mark, determined by the point in time when the mark() method was invoked.
    - Few input streams support this functionality, and may cause an IOException to be thrown if called.



## The java.io.InputStream Class (Cont.)

- The abstract InputStream class defines methods common to all input streams and all of them are public:

⑨ – long skip (long amount) throws java.io.IOException

- Reads, but ignores, the specified amount of bytes.
- These bytes are discarded, and the position of the input stream is updated.
- Though unlikely, it is entirely possible that the specified number of bytes could not be skipped (for example, as stated in the Java API, if the end of the stream is reached).
- The skip method returns the number of bytes skipped over, which may be less than the requested amount.

بدي اقل skip لعدد من bytes بين ما اقلهم !

عدد الذي تحمله skip

بتتج

قديه

بالفعل

skip

بلا يمكن

يوجد ال end

قبل ما يوصل

كل ال skip

لأنه يكون أقل

أو يساوي الذي طلبته ياه



\* الأفضل تحطى ال File بنفسى Path البروجكت  
عشان تكتبى اسمه بين جردون ال Path كامل .

← بنا نقرأ من File وبنطالعهم عالشاشة .

## Example Using a Low-level Input Stream

```

import java.io.*;

public class FileInputStreamDemo {
    public static void main(String[] args) {
        if (args.length != 1){
            System.err.println ("Syntax - FileInputStreamDemofile");
            return;
        }
        try{
            InputStream fileInput = new FileInputStream( args[0] );
            int data = fileInput.read();
            while (data != -1){
                System.out.write ( data );
                data = fileInput.read();
            }
            fileInput.close();
        } catch (IOException ioe){
            System.err.println ("I/O error - " + ioe);
        }
    }
}

```

\* write → بتطبع characters  
\* Println → بتطبع ال integers

بتأكد إنه عندي File واحد بين إيدي  
مقول من واحد

لو كان ال File خاصي

System.out.Flush  
له بنحتاج نكتبها في بلدا

ال Versions  
عشان اتجيب

ال Data من ال buffer اللي بال  
System.out

نقرأ أول byte بال data →  
بعدين بنصين نقرأ حرفي حرف  
لأنه ال File بالانجليزي  
والأحرف الإنجليزية بتأخذ  
1 byte ، لو عربية كان كل حرفين  
يقرأ حرف .

\* الحروف الإنجليزية لينعلم encoding بتحتاجوا 1 byte  
والعربية 2 bytes  
\* في أنواع بتطوي الكل 2 bytes  
by default وفي أنواع لا .

Eng. Asma Abdel Karim  
Computer Engineering Department

## Writing to an Output Stream

• While an input stream is a data consumer, an output stream is a data producer.

\*— It literally creates bytes of information and transmits them to something else (such as a file or data structure or network connection).

• بنفسى  
الترتيب اللي  
بتكتب فيه يروح  
تطلع ال Data

Like input streams, data is communicated sequentially that is, the first byte in will be the first byte out.

– This approach is analogous to a FIFO queue.

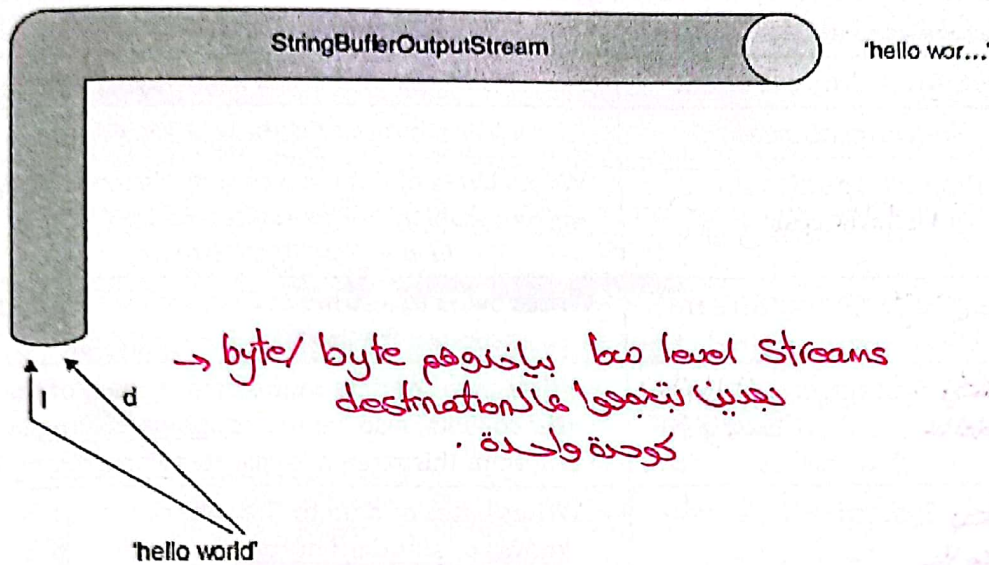
• Unlike some specialized filter input streams, which allow you to "go back n" bytes within a sequence, once data is sent to an output stream it cannot be undone.



Eng. Asma Abdel Karim  
Computer Engineering Department

له مافي . reset و Mark  
بالكتابة +  
أول ما ال data تنكتب  
ما يصح تعطل undo .

## Writing to an Output Stream (Cont.)



Eng. Asma Abdel Karim  
Computer Engineering Department

25

## Writing to an Output Stream (Cont.)

- A number of output streams are available in the java.io package for a variety of tasks.
  - \* – Such as writing to data structures including strings and arrays, or to files or communication pipes.
- There are six important low-level output streams that may be written to.
  - \* – In addition to filter streams that may be connected to these low-level streams
- As mentioned earlier, there are other streams which may be written to that developers cannot create and instantiate directly. → like socket



Eng. Asma Abdel Karim  
Computer Engineering Department

26

# Writing to an Output Stream (Cont.)

"حسنتخدمه بالopp"

Low-level Output Stream	Purpose of Stream
ByteArrayOutputStream	Writes bytes of data to an array of bytes.
FileOutputStream	Writes bytes of data to a local file.
PipedOutputStream Multithreading	Writes bytes of data to a communications pipe, which will be connected to a <code>java.io.PipedInputStream</code> .
StringBufferOutputStream	Writes bytes to a string buffer (a substitute data structure for the fixed-length string).
System.err Console بطلع باللون الأحمر عادة	Writes bytes of data to the error stream of the user console, also known as <u>standard error</u> . In addition, this stream is <u>cast to a <code>PrintStream</code></u> .
System.out Console بكتب	Writes bytes of data to the user console, also known as <u>standard output</u> . In addition, this stream is <u>cast to a <code>PrintStream</code></u> .

الوهم فوائد  
لو ركبنا  
Filter Streams  
لو طبقت  
علاقتنا بس  
ممكن يتخربش  
ترتيب الطباعة



Eng. Asma Abdel Karim  
Computer Engineering Department

ما خبير صفقاتهم  
PrintStream

Methods & Print In  
Filter stream  
Print In  
PrintStream Casting  
low level

# Writing to an Output Stream (Cont.)

- Bytes may be sent one at a time or as part of an array.
- However, when bytes are read one at a time, individual byte writes may affect system performance. Reading information can block indefinitely, but writing information may also block for small amounts of time.
- This is not normally as significant an issue as the case of blocking read operations, as the bytes are ready to send.

عملية القراءة  
blocking  
عملية الكتابة  
بتأخر  
Performance  
أكثر



Eng. Asma Abdel Karim  
Computer Engineering Department

# The java.io.OutputStream Class

The abstract class java.io.OutputStream defines the following public methods:

– void close() throws java.io.IOException

1. Closes the output stream, notifying the other side that the stream has ended. → يعني تنبيه انه هذا ال File سكر عن طريق ارسال ١- للFiles التي بتقرأ من هذا ال File
2. Pending data that has not yet been sent will be sent, but no more data will be delivered.

لو كان في buffer ممكن في data تكون معلقة بال buffer فلما أسكر ال Stream بنعملوا Flush عشان تطلع ، بس ما يقدر أكتب على ال File صدام سكرته.



## The java.io.OutputStream Class (Cont.)

The abstract class java.io.OutputStream defines the following public methods: \*مرات لو بنسى عمل flush ممكن البرنامج يعلق

– void flush() throws java.io.IOException

1. Performs a "flush" of any unsent data and sends it to the recipient of the output stream.
2. To improve performance, streams will often be buffered, so data remains unsent. This is useful at times, but obstructive at others.
3. The method is particularly important for OutputStream subclasses that represent network operations, as flushing should always occur after a request or response is sent so that the remote side isn't left waiting for data. → شرحها ب Slide 55



# The java.io.OutputStream Class (Cont.)

The abstract class java.io.OutputStream defines the following public methods:

– void write (int byte) throws java.io.IOException

1. Writes the specified byte.
2. This is an abstract method, overridden by OutputStream subclasses.

class ←  
override  
بالطريقة الاب  
تناسبه.

– void write (byte[] byteArray) throws java.io.IOException

1. Writes the contents of the byte array to the output stream.
2. The entire contents of the array (barring any error) will be written.



Eng. Asma Abdel Karim  
Computer Engineering Department

31

# The java.io.OutputStream Class (Cont.)

The abstract class java.io.OutputStream defines the following public methods:

– void write (byte[] byteArray, int offset, int length) throws java.io.IOException

1. Writes the contents of a subset of the byte array to the output stream.
2. This method allows developers to specify just how much of an array is sent, and which part, as opposed to the OutputStream.write(byte[] byteArray) method, which sends the entire contents of an array.

دكتب ←  
جزء من  
Array

مفروض ←



Eng. Asma Abdel Karim  
Computer Engineering Department

32

2020



# Example Using a Low-level Output Stream

```
import java.io.*;
```

ليقرأ من File ويكتب على File  
• 2 args

```
public class FileOutputStreamDemo{  
    public static void main(String args[]){  
        if (args.length != 2){  
            System.err.println("Syntax - FileOutputStreamDemo src dest");  
            return;  
        }  
        String source = args[0];  
        String destination = args[1];  
        try {  
            InputStream input = new FileInputStream( source );  
            System.out.println ("Opened " +source + " for reading.");  
            OutputStream output = new FileOutputStream( destination );  
            System.out.println ("Opened " +destination + " for writing.");
```

→ FileNotFoundException exception

\* لما أحاول افتح File و File Input Stream  
! لما أحاول افتح أو أركب File output Stream  
• error ما يسهي error



Eng. Asma Abdel Karim  
Computer Engineering Department

ليطبع ال ASCII (أرقام)

التعامل مع شكل bytes

## Example Using a Low-level Output Stream (Cont.)

```
        int data = input.read();  
        while ( data != -1){  
            output.write (data);  
            data=input.read();  
        }  
        input.close();  
        output.close();  
        System.out.println ("I/O streams closed");  
    }  
    catch (IOException ioe){  
        System.err.println ("I/O error - " + ioe);  
    }  
}
```



Eng. Asma Abdel Karim  
Computer Engineering Department

34

② أفضل للـ Processing و فيرو buffering ①

# Filter Streams

\* نوع الـ data الذي يتعامل معها مش bytes ممكن تكون أحرف كزي أو إنجليزي، توأرقام...

فقرعة الـ data من طريق الـ low level stream = بالـ bytes مقلقة وغير مجدية

While the basic low-level streams provide a simple mechanism to read and write bytes of information their flexibility is limited.

After all, reading bytes is complex.

- \* - There's more to the world than just bytes of data.
- \* - Text, for example, is a sequence of characters, and other forms of data like numbers take up more than a single byte.

Byte-level communication can also be inefficient.

- Data buffering can improve performance.
- To overcome these limitations, filter streams are used.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Filter Streams (Cont.)

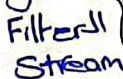
Filter streams add additional functionality to an existing stream.

- ① - By processing data in some form, such as buffering for performance.
- ② - By offering additional methods that allow data to be accessed a different manner. For example, reading a line of text rather than a sequence of bytes.

Filters make life easier for programmers.

الـ filter stream ما بقدر أنا أحول لـ bytes وهيك ، بحس اكتب واقف بنفس نوعهم وهو يتحولهم لـ bytes

- ① - As they can work with familiar constructs such as strings, lines of text, and numbers, rather than individual bytes.
- ② - Instead of the programmer writing a string one character at a time and converting each character to an int value for the OutputStream.write(int) method, the filter stream does this for them.



Eng. Asma Abdel Karim  
Computer Engineering Department

\* ما في غير الـ low-level stream لازم يكون دايمًا أول خطوة هو

# Connecting a Filter Stream to an Existing Stream

\* ال low-level ما يركب عليه كمان ال level. صا.  
 \* ال Filter يركب ال low-level بعدين  
 بيس يركبوا أكثر من ال Filter stream بعض.

- \* Filter streams can be connected to any other stream.
  - \* - To a low-level stream or even another filter stream.
- \* Filter streams are extended from the java.io.FilterInputStream and java.io.FilterOutputStream classes.
- \* Each filter stream supports one or more constructors That accept:
  - Either an InputStream, in the case of an input filter, or
  - An OutputStream, in the case of an output filter.
- Connecting a filter stream is as simple as:
  - \* - Creating a new instance of a filter passing an instance of an existing stream, and
  - \* - Using the filter from then on to read or write.

لازم عالأقل parameter 1  
 يكون يا إما input stream  
 ال التي تنكب عليه أو output stream  
 ال التي تنكب عليه



# Connecting a Filter Stream to an Existing Stream (Cont.)

- The following code connects a PrintStream (used to print text to an OutputStream subclass) to a stream that wrote to a file and uses the filter stream to write a message on the file.

```

FileOutputStream fout = new FileOutputStream ( somefile );
PrintStream pout = new PrintStream (fout);
pout.println ("hello world");
    
```

ال الذي يركب ال low-level  
 بعض ال اسم ال File  
 ال ال system.out وال system.error  
 ما تخين منه ال Print وال Println  
 ال level ما يركب عليه ال Filter  
 ال ال level ال الذي يركب عليه  
 موجودين فيه ال Print وال Println  
 ممكنك تكتب ال data و text  
 \* بتتبع ال hello world ال File ال Pout  
 هو ال الذي بتحول منه  
 نوع ال الذي تكتبه ل bytes  
 تكتب ال File



# Connecting a Filter Stream to an Existing Stream (Cont.)

مدمجت Filter

• فالمنطق تستخدم لتعيين القراءة والكتابة

The process is fairly simple as long as the programmer remembers two things:

1. Read and write operations must take place on the new filter stream.
2. Read and write operations on the underlying stream can still take place, but not at the same time as an operation on the filter stream.

ما في مشكلة انك تستخدم "out" في المثال السابق بس ما تقمل override لانه من المنوع ال out out ما يطع زي overlapping ما بديك

\* نفس اول ال level access ال address تاخو الا في حالة وحدة:

```
FileOutputStream out = new FileOutputStream("dst.txt");
PrintStream out = new PrintStream(out);
```

access low level (outputstream)

Eng. Asma Abdel Karim  
Computer Engineering Department

PrintStream FilterStream جوا out لانه صاير ياشر عال

## Useful Filter Input Streams

### BufferedInputStream Class

- The purpose of I/O buffering is to improve system performance.
- Rather than reading a byte at a time, a large number of bytes are read together the first time the read() method is invoked.
- When an attempt is made to read subsequent bytes, they are taken from the buffer, not the underlying input stream.

\* - This improves data access time and can reduce the number of times an application blocks for input.

ال access عليه وال بال local ال البرنامج تبعك فقتيب عليه ال buffer ما ال داعي



Eng. Asma Abdel Karim  
Computer Engineering Department

# Useful Filter Input Streams

## BufferedInputStream Class (Cont.)

- Constructors:
  - \* – `BufferedInputStream (InputStream input)`  
Creates a buffered stream that will read from the specified `InputStream` object.  
*بأخذ أي instance من النوع Filter stream InputStream*
  - \* – `BufferedInputStream (InputStream input, int bufferSize)`  
throws `java.lang.IllegalArgumentException`  
*لو بيحدد الsize للbuffer ، لازم يكون > 0 وموسلوب*
    1. Creates a buffered stream, of the specified size, which reads from the `InputStream` object passed as a parameter.
    2. This allows developers to specify a size, which can improve efficiency if large amounts of data are going to be read. The buffer size specified must be greater than or equal to one.



# Useful Filter Input Streams

## BufferedInputStream Class (Cont.)

- Methods:
  - \* – No additional methods are provided by the `BufferedInputStream` class.  
*ما يعطيك Method إضافية بس ال data بتقعد فيه ولما تبجي تقدر بتكون ال data بال buffer بتسأل*
  - \* – However, it does override the `markSupported()` method, indicating that it supports the `mark(int)` and `reset()` methods.



## ② Useful Filter Input Streams

### DataInputStream Class

Formating ال ←  
بدلما اقرأ bytes بقى بالنوع  
اللي موجود  
قسي  
وهي بتكوني  
ياهم ل bytes

- A frequent task in any programming language is reading and writing primitive data types such as numbers and characters.
- \* – These information types are not easily represented as bytes (for example, some data types take up more than one byte of information).
- Developers should not be concerned with the way in which representation occurs.
- \*• Instead, the data types can be read simply, by invoking methods of the DataInputStream class, which handles the translation automatically.
- This class implements the java.io. DataInput interface.

هامة



Eng. Asma Abdel Karim  
Computer Engineering Department

43

## Useful Filter Input Streams DataInputStream Class (Cont.)

- Constructors:
  - DataInputStream (InputStream input)  
Creates a data input stream, reading from the specified input stream.

← اسم ال input stream الرئيسي الزمعة

Small not capital  
Primitive data type

## Useful Filter Input Streams DataInputStream Class (Cont.)

### • Methods:

Many methods are added to the DataInputStream class, in order to facilitate access to new data types.

- **boolean** readBoolean( ) throws java.io.EOFException java.io
- **byte** readByte( ) throws java.io.EOFException java.io.IOException
- **char** readChar( ) throws java.io.EOFException java.io.IOException
- **double** readDouble( ) throws java.io.EOFException java.io.IOException
- **float** readFloat( ) throws java.io.EOFException java.io.IOException
- **void** readFully(byte[ ] byteArray) throws java.io.EOFException java.io.IOException
- **void** readFully(byte[] byteArray, int offset, int length) throws java.io.EOFException java.io.IOException

بشيء  
الذي ياد  
اعلا-سا  
له بلا لها بشي لو بنجوم اختلاف



Eng. Asma Abdel Karim  
Computer Engineering Department

→ End of file (stream)

45

\* كيف نعرف انه وصلنا لا end of stream ؟ يعني EOFException لانهم ما يسلموا الـ زي

## Useful Filter Input Streams

ال level-سا

## DataInputStream Class (Cont.)

### • Methods:

- **int** readInt( ) throws java.io.EOFException java.io.IOException
- **String** readLine( ) throws java.io.IOException
- **long** readLong( ) throws java.io.EOFException java.io.IOException
- **short** readShort( ) throws java.io.EOFException java.io.IOException
- **int** readUnsignedByte( ) throws java.io.EOFException java.io.IOException
- **int** readUnsignedShort( ) throws java.io.EOFException java.io.IOException
- **String** readUTF( ) throws java.io.EOFException java.io.IOException
- **Static String** readUTF(DataInputStream input) throws java.io.EOFException java.io.IOException
- **Int** skipBytes(int number) throws java.io.IOException

هدفع ابي اقدر اقرأ ال Data بأنواع  
أعلى من ال byte

بنوع لما يصيب خطأ في القراءة

بلا int لأنه بال byte  
ما في unsigned

→ skip number of bytes



Eng. Asma Abdel Karim  
Computer Engineering Department

46

مخصصا لما بي اقول lines من text ، بعد عدد الـ lines .  
أقول واحد يعني Useful Filter Input Streams

## ③ LineNumberInputStream Class

- This class provides helpful functionality by tracking the number of lines read from an input stream. → مش موجود
- It is deprecated as of JDK1.1, however, since the preferred way to process text data is to use a reader class. ← لقناهم شرحهم (اي شئ انا علاقة بلا text)
- Also, line numbers are not very serviceable in terms of a stream of bytes.
- Nonetheless, if writing for JDK1.02 systems, it may be useful. → ممكن يكون مش موجود



Eng. Asma Abdel Karim  
Computer Engineering Department

41

## Useful Filter Input Streams LineNumberInputStream Class (Cont.)

- Constructors:  
– `LineNumberInputStream(InputStream input)`  
Creates a line number stream, reading from the specified input stream. → Instance الجيرب عليه
- Methods:
  - ① – `int getLineNumber()`  
✳ Returns the number of lines that have been read by this input stream.
  - ② – `void setLineNumber(int number)`  
✳ Modifies the line number counter to the specified value.



Eng. Asma Abdel Karim  
Computer Engineering Department

42



# Useful Filter Input Streams

الهدف منه إنه  
الفعال Data  
أكثر من مرة، في  
buffer أكيد.

٥

## PushBackInputStream Class

- The PushBackInputStream class allows a single byte to be read and then "pushed back" into the stream for later reading.
- An internal buffer is maintained that allows data to be pushed back into the front of the input stream buffer, or added if the data had never been read from it.
- This is useful when the programmer needs to take a "sneak peek" at what's coming.

– For example in a text parser or to determine what the next command in a communications protocol is going to be.

لو بي اقرأ البيانات أكثر من مرة وأخبرها.

أيدي  
النظر بال  
البيانات وأشوف  
شئ وحدها  
فستعمل  
Pushback



Eng. Asma Abdel Karim  
Computer Engineering Department

\* ما بيقرأ كثير لأنه بيقرأ باستخدام ال mark وال reset  
بيقال بيقرأ عن ال mark و reset إنهما ممكن  
تخليج بيقرأ جديدة عنها ما انقرات بال buffer

# Useful Filter Input Streams

## PushBackInputStream Class (Cont.)

- Constructors:

`PushBackInputStream(InputStream input)`

Creates a PushBackInputStream that will read from the specified input stream.

`PushBackInputStream (InputStream input int bufferSize)` throws java.lang.IllegalArgumentException

1. Creates a PushBackInputStream that will read from an input stream and use a buffer of the specified size.
2. If a value of less than one is specified for the buffer size, an exception will be thrown.



Eng. Asma Abdel Karim  
Computer Engineering Department

50

# Useful Filter Input Streams

## PushBackInputStream Class (Cont.)

- Methods:

- \* – void **unread** (byte[] byteArray) throws java.io.IOException  
Pushes back the contents of the specified array. If a buffer overrun occurs, an exception is thrown.

- \* – void **unread** (byte[] byteArray, int offset, int length) throws java.io.IOException

Pushes back a subset of the contents of the specified array, starting at the specified offset and lasting for the specified duration. If a buffer overrun occurs, an exception is thrown.

- \* – void **unread** (int byte) throws java.io.IOException

Pushes back the specified byte into the front of the buffer. If a buffer overrun occurs, an exception is thrown.

← لو ال buffer ماز full يترجي error



Eng. Asma Abdel Karim  
Computer Engineering Department

51

# Useful Filter Output Streams

## BufferedOutputStream Class

- The BufferedOutputStream provides data buffering similar to the BufferedInputStream.

- As suggested by the name of the class, however, it buffers writes, not reads.

- An internal buffer is maintained, and when the buffer is complete or if a request to flush the buffer is made, the buffer contents are dumped to the output stream to which the buffered stream is connected.



\* از صيد 30 Slide \*



Eng. Asma Abdel Karim  
Computer Engineering Department

لـ سب تكتب ال data ما يتطلع يترجي  
ال Stream لا يتجمع بال buffer اول

\* فادته ايز بعض  
ال bandwidth وال سرعة  
سب اول في حال بيبي  
تطلع ال data  
52  
قبل ما يتبع ال buffer كامل وانزل

الشي ليستخدم ال flush

# Useful Filter Output Streams

## *BufferedOutputStream* Class (Cont.)

- Constructors:

== *BufferedOutputStream* (*OutputStream* output)

1. Creates a buffer for writing to the specified output stream.

2. The default size of this buffer is 512 bytes in length.

== *BufferedOutputStream* (*OutputStream* output int bufferSize) throws *java.lang.IllegalArgumentException*

1. Creates a buffer for writing to the specified output stream, overriding the default buffer sizing.

2. The buffer is set to the specified buffer size, which must be greater than zero or an exception is thrown.



# Useful Filter Output Streams

## *BufferedOutputStream* Class (Cont.)

- Methods:

\* – No extra methods have been added to this class.

\* – However, the *flush()* method has been overridden.

== • It will flush the contents of a buffer, sending it immediately to the output stream it is connected to.

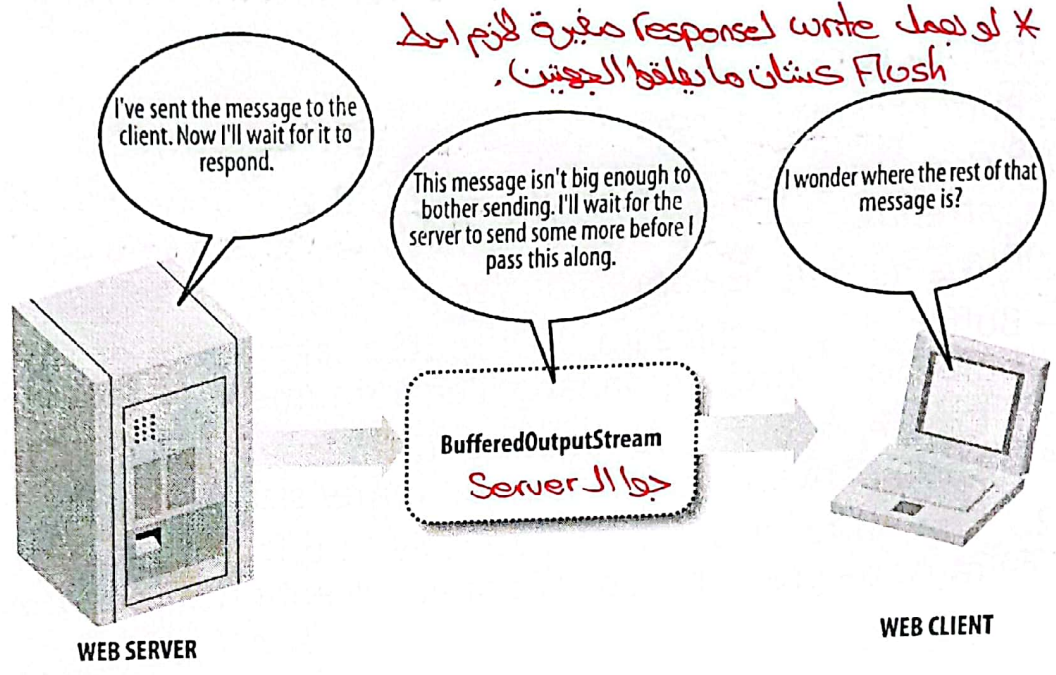
== • This is particularly important in networking, as a protocol request can't be sent if it is still stuck in the buffer, and the remote program may be waiting for a response.



\* ال request عادة بتكون اشئ صغير ، فلو انبعت وكلفنا ليقتل ال  
 buffer رح يعلق البرنامج من الجويتين لو ما استخدمنا Flush .  
 \* ال response أيضا عادة بتكون اشئ صغير .

# Useful Filter Output Streams

## BufferedOutputStream Class (Cont.)



\* لو عمل write response صغيرة لازم انا  
 Flush كشان ما يعلق الجويتين .

# Useful Filter Output Streams

← بيخفي افي اكتب  
 ال data بـ Formatting  
 مختلف

## DataOutputStream Class

- Like the DataInputStream class, the DataOutputStream class is designed to deal with primitive datatypes, such as numbers or bytes. → ال data اللي بتاخذ أكثر من byte
- Most of the read methods of DataInputStream have a corresponding write method mirrored in DataOutputStream.
- \* - This allows developers to write datatypes to a file or other type of stream, and to have them read back by another Java application without any compatibility issues over how primitive datatypes are represented by different hardware and software platforms.
- It implements the java.io.DataOutput interface, which provides additional methods for writing primitive datatypes.

\*

\* طريقة القراءة والكتابة لانهم تكون متناظرة  
 يعني لو كتبت int بيقرأ int بـ  
 وهكذا .

# Useful Filter Output Streams

## DataOutputStream Class (Cont.)

- Constructors:

← ال Instance التي يدك تركيبها عليه

- `DataOutputStream (OutputStream output)`

*Creates a data output stream, which will write to the specified stream.*



# Useful Filter Output Streams

## DataOutputStream Class (Cont.)

- Methods:

مثلا موجودة بالمعنى → عدد بعد عدد ال bytes التي صرت كتابتها

- \* – `int size( )` Returns the number of bytes written to the data output stream.

- \* – `void writeBoolean (boolean value)` throws `java.io.IOException`  
Writes the specified boolean value, represented as a one-byte value.

- \* – `void writeByte (int byte)` throws `java.io.IOException`  
Writes the specified byte to the output stream.

- \* – `void writeBytes (String string)` throws `java.io.IOException`  
Writes the entire contents of a string to the output stream a byte at a time.

- \* – `void writeChar (int char)` throws `java.io.IOException`  
Writes the character to the output stream as a two-byte value.

- \* – `void writeChars (String string)` throws `java.io.IOException`  
Writes the entire contents of a string to the output stream, represented as two-byte values

شئ الفرق بينهم؟



# Useful Filter Output Streams

## DataOutputStream Class (Cont.)

- Methods:
  - \* – void writeDouble (double doubleValue) throws java.io.IOException  
Converts the specified double value to a long value and then converts it to an eight-byte value.
  - \* – void writeFloat (float floatValue) throws java.io.IOException  
Converts the specified float value to an int and then writes it as a four-byte value.
  - \* – void writeInt (int intValue) throws java.io.IOException  
Writes an int value as a four-byte value.
  - \* – void writeLong (int intValue) throws java.io.IOException  
Writes a long value as eight bytes.
  - \* – void writeShort (int intValue) throws java.io.IOException  
Writes a short value as two bytes.
  - \* – void writeUTF (String string) throws java.io.IOException  
Writes a string using UTF-8 encoding.



Eng. Asma Abdel Karim  
Computer Engineering Department

59

# Useful Filter Output Streams

3) **PrintStream Class** → Filter Stream  
 ← إضافي مش موجود بال default  
 ، بيكتا نكتب ال data ك text

- The PrintStream is the most unusual of all filter output streams.   
 ← ما برهوي exception بيحفظ انه   
 ← صار
- \* – It is atypical in that it overrides methods inherited from FilterOutputStream without throwing the expected java.io.IOException class.
- The PrintStream adds additional methods as well.
  - \* – None of which may throw an IOException.
  - \* – No errors are overtly reported, and instead the presence of an error is determined by invoking the checkError() method—although no further details may be obtained as to the cause of the error.
- Despite its idiosyncrasies, the PrintStream is an extremely useful class.   
 ← رفض النطوع فيه مختلفاى الباقي  
 ←
- \* – It provides a convenient way to print primitive datatypes as text using the print(..) method, and to print these with line separators using the println(..) method.



Eng. Asma Abdel Karim  
Computer Engineering Department

60

نشرها اخطيه

بكتب ك (text)

يطوع زي  
ماهو  
لنطوع

# Useful Filter Output Streams

## PrintStream Class (Cont.)

### Constructors:

- PrintStream (OutputStream output)

Creates a print stream for printing of datatypes as text.

- PrintStream (OutputStream output, boolean flush)

1. Creates a print stream for printing of datatypes as text.
2. If the specified boolean flag is set to "true," whenever a byte array, println method, or newline character is sent, the underlying buffer will be automatically flushed.

System.out و System.err ←

لا يوجد buffering



Eng. Asma Abdel Karim  
Computer Engineering Department

61

# Useful Filter Output Streams

## PrintStream Class (Cont.)

### Methods:

- boolean checkError()

Automatically flushes the output stream and checks to see if an error has occurred. Instead of throwing an IOException, an internal flag is maintained that checks for errors.

- \* - void print (boolean value)  
Prints a boolean value.
- \* - void print (char character)  
Prints a character value.
- \* - void print (char[] charArray)  
Prints an array of characters.
- \* - void print (double doubleValue)  
Prints a double value.
- \* - void print (float floatValue)  
Prints a float value.

بما ان يكون true لو لا يرجع False  
ان كان صلي error من آخر عمليات صليت  
نحرف أي بيانات أخرى.



Eng. Asma Abdel Karim  
Computer Engineering Department

62

PrintStream \* buffered Stream  
منه يحتاج  
flush  
يمكن  
دمج  
حساب  
العمل  
Flush  
الغرض للاستيلاء

زي ما هم علا File > text

# Useful Filter Output Streams

## PrintStream Class (Cont.)

### • Methods:

- void print (int intValue)  
Prints an int value.
- void print (long longValue)  
Prints a long value. → بتطبع الـ toString()
- void print (Object obj)  
Prints the value of the specified object's toString() method.
- void print (String string)  
Prints a string's contents.
- void println() → new line بعده  
Sends a line separator (such as '\n'). This value is system dependent and determined by the value of the system property "line.separator."
- void println (char character)  
Prints a character value, followed by a println().
- void println (char[] charArray)  
Prints an array of characters, followed by a println().



Eng. Asma Abdel Karim  
Computer Engineering Department

63

# Useful Filter Output Streams

## PrintStream Class (Cont.)

### • Methods:

- void println (double doubleValue)  
Prints a double value, followed by a println().
- void println (float floatValue)  
Prints a float value, followed by a println().
- void println (int intValue)  
Prints an int value, followed by a println().
- void println (long longValue)  
Prints a long value, followed by a println().
- void println (Object obj)  
Prints the specified object's toString() method, followed by a println().
- void println (String string)  
Prints a string followed by a line separator.
- protected void setError()  
Modifies the error flag to a value of "true."



Eng. Asma Abdel Karim  
Computer Engineering Department

64



# References

**Chapter 4** of *Java™ Network Programming and Distributed Computing*, David Reilly and Michael Reilly.



Eng. Asma Abdel Karim  
Computer Engineering Department

65

# Networks and Internet Programming

Data Streams

Part-II



Eng. Asma Abdel Karim  
Computer Engineering Department

1

# Outline

- Readers and Writers. → عنوان "text" تم إنشاؤه
- Object Persistence and Object Serialization.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Overview

- While input streams and output streams may be used to read and write text as well as bytes of information and primitive data types, a better alternative is to use readers and writers.
- Readers and writers were introduced in JDK1.1 to better support Unicode character streams.

التي هي عبارة عن Streams  
منها مصنفين عن التي قبل



Eng. Asma Abdel Karim  
Computer Engineering Department



بالنسبة، التي هي عبارة عن  
Stream of char.  
منها bytes زي قبل

# What Are Unicode Characters?

- Most people think of characters as being composed of 8 bits of data, offering a range of 256 possible characters.
  - \* – Low ASCII (0–127) characters are followed by high ASCII characters (128–255).
  - \* – The high ASCII characters represent characters and symbols such as those used in foreign languages or punctuation.
- However, people quickly realized that even 256 characters were not enough to handle the many characters used in languages around the world. This is where Unicode came in.



Eng. Asma Abdel Karim  
Computer Engineering Department

4

## What Are Unicode Characters? (Cont.)

- Unicode characters are represented by 16 bits.
  - Allowing for a maximum of 65,536 possible characters.
- Unicode characters are supported by Java.
- Java also supports a modified form called UTF-8.
  - This is a variable-width encoding format; some characters are a single byte and others multiple bytes.

هو ال Standard أو ال default بال Java →  
بسن ال default  
كوال unicode  
وبيك تحويل  
UTF-8 أو J  
نوع تافا بيوطي ال char ال عدد اللي يحتاجه من ال bytes فتة →  
غير لازم  
تكتب code  
لتحولهم.



Eng. Asma Abdel Karim  
Computer Engineering Department

5

# The Importance of Readers and Writers

- لو يتعامل مع أرقام الأفضل استخدم الـ *DataStream*
- For those dealing solely with primitive data types, use of input streams and output streams may by all means be continued.
  - However, if applications are processing text information only, use of a reader and/or a writer, to better support Unicode characters, should be considered.
- أما لو كانت الـ *Data* عبارة عن *Text* فقط فالـ *Readers*, *writers* أفضل كمنهجية وخيره.



Eng. Asma Abdel Karim  
Computer Engineering Department

6

## التحويل بين *DataStreams*, *Readers/writers* والعكس يكون سهواً جداً

### From Input Streams to Readers

الـ *DataStream* الـ *Super*

- The *java.io.InputStream* class has a character-based equivalent in the form of the *java.io.Reader* class.
- The reader class has similar method signatures to that of the *InputStream* class. *للتبسيط الكثير الفرق بسيط*
  - Existing code may be quickly converted to use it.

الاختلاف بينهم

- However:

- ① – Some slight changes are made to the method signatures, to support character, and not byte, reading.
- ② – The *available()* method has been removed, and replaced by the *ready()* method.

لأن ما يخرج عدد الـ *bytes* التي صالحة

النتيجة T/F بس.



Eng. Asma Abdel Karim  
Computer Engineering Department

7

# The java.io.Reader Class

- Constructors:

No public constructors are available for this class. Instead, a reader subclass should be instantiated.

\* زي ما كنا سيني low level I/O streams بقراو و بكتبو bytes و  
high level I/O streams (Filter) بكتبو وبقراو ب Format آلى من byte وبعملو buffering



Eng. Asma Abdel Karim  
Computer Engineering Department

برخه في سيني low level Readers/writes بقرا

و بكتبو character سيني high level Readers/writes  
(Filters) بكتبو وبقراو ب Format آلى زي آفرا line كامل بديل char منفصلين.

## The java.io.Reader Class (Cont.)

\* برخه بفتح فيهم buffers وبعملو

- Methods:

The class includes the following methods, all of which are public:

– void close() throws java.io.IOException

Closes the reader. → قديو سيني بقرا char قبل ما ال mark تضيع و تفتح لأول اشئي.

– void mark(int amount) throws java.io.IOException

1. Marks the current position within the reader, and uses the specified amount of characters as a buffer.

2. Not every reader will support the mark(int) and reset() methods.

– boolean markSupported() → default يكون False

Returns "true" if the reader supports mark and reset operations.



Eng. Asma Abdel Karim  
Computer Engineering Department

\* مين Readers يكون بفتح  
ال Mark/reset بفتح

9

# The java.io.Reader Class (Cont.)

## • Methods:

The class includes the following methods, all of which are public:

- **int read()** throws java.io.IOException  
Reads and returns a character, blocking if no character is yet available. If the end of the reader's stream has been reached, a value of -1 is returned.  
*تقرأ 1 char. ← تنقل من byte تنقل عند الحجم أو قد الموجود بال Reader →*
- **int read(char[] characterArray)** throws java.io.IOException  
Populates an array of characters with data. This method returns an int value, representing the number of bytes that were read. If the end of the reader's stream is reached, a value of -1 is returned and the array is not modified.  
*تقرأ 2 byte أو حتى least 2 byte التي فيهم ال data → Reader → not blocking ما يظننا بال حدود ال array لان*
- **int read(char[] characterArray, int offset, int length)** throws java.io.IOException  
Populates a subset of the array with data, starting at the specified offset and lasting for the specified duration. This method returns an int value, representing the number of bytes read, or -1 if no bytes could be obtained.  
*ال array لان*



Eng. Asma Abdel Karim  
Computer Engineering Department

10

# The java.io.Reader Class (Cont.)

## • Methods:

The class includes the following methods, all of which are public:

- **boolean ready()** throws java.io.IOException  
Returns "true" if there is data available, or "false" if not. This is similar to the `InputStream.available()` method, except that the number of bytes/characters is not available.
- **void reset()** throws java.io.IOException  
Attempts to reset the reader's stream, by moving back to an earlier position. Not every reader supports either mark or reset, and an exception could be thrown or the request ignored.  
*لبرجوني ← للمكان التي تركت في ال mark*
- **long skip(long amount)** throws java.io.IOException  
Reads and discards the specified number of characters, unless the end of the input stream is reached or another error occurs. The skip method returns the number of characters successfully skipped.  
*قديه يعمل ← Skip*



Eng. Asma Abdel Karim  
Computer Engineering Department

ممكن ما يعمل Skip للعدد كالم يعمل

لعدد أقل من ال long amount  
لأسباب مختلفة، بس مستحيل  
يعمل Skip للعدد أكبر

11

# The java.io.Reader Class (Cont.)

- Like input streams: "أول شيء في Stop" →
  - There are a variety of low-level readers (which connect to a data source, such as a file or a data structure), and
  - There are filter readers for high-level communication tasks.



Eng. Asma Abdel Karim  
Computer Engineering Department

12

## Low-Level Readers

### ① CharArrayReader Class

- The CharArrayReader class is a reader that obtains data by reading characters from an *array*.
- Constructors:
  - CharArrayReader(char[] charArray)  
*Creates a character array reader that will operate on the specified array.*
  - CharArrayReader(char[] charArray, int offset, int length)  
*Creates a character array reader that will operate only on a subset of the specified array, starting at the specified offset and lasting for the specified length.*
- Methods:

The CharArrayReader adds no new methods.

→ بحسبهم لما تتركب ال Filter Readers ونقدر بـ Format أولي



Eng. Asma Abdel Karim  
Computer Engineering Department

13

عكس الـ output File عادي لو ما كان في File هو يشوع الـ File ويكتب عليه

## Low-Level Readers

### ② FileReader Class

لانم اناكد انه الـ File موجود بالقراءة.

- This reader obtains its data directly from a local file, similar to the `FileInputStream` class.
- Care must be taken, as with the `FileInputStream` class, when creating an instance of it, as an exception will be thrown:

- If the file could not be located, or  
لو ما في File
- If security access permissions restrict it from being read.  
لو ما مسموح نفوت نقرأ الـ File



Eng. Asma Abdel Karim  
Computer Engineering Department

## Low-Level Readers FileReader Class (Cont.)

- Constructors:  $\rightarrow$  object
  - `FileReader(File file)` throws `java.io.FileNotFoundException`  
Creates a reader that will access the contents of the specified file object, if the file it represents exists.
  - `FileReader(String filename)` throws `java.io.FileNotFoundException`  
Creates a reader that will access the contents of the specified filename, if it exists.  $\rightarrow$  name
  - `FileReader(FileDescriptor descriptor)`  
Creates a reader that will access the contents of the specified descriptor handle.  $\rightarrow$  object  $\rightarrow$  File بأش
- Methods: The `FileReader` class adds no new methods.



Eng. Asma Abdel Karim  
Computer Engineering Department





## Low-Level Readers

القسم بالثلاثة threads

### ③ PipedReader Class

- Constructors:

- **PipedReader()**

Creates an unconnected pipe reader.

\* ال Piped من الجيت لازم أنتبه  
• بعض

- **PipedReader(PipedWriter writer)**

Creates a connected pipe that will read the output of the specified writer.

- Methods:

A single (public) method is added by this class.

- **void connect(PipedWriter writer)** throws java.io.IOException

Connects the reader to the specified writer. Any output that is sent by the piped writer may then be read by the piped reader.

(كل شئ يكتبه على writer يوصل ال reader اللي بقراء منه)



Eng. Asma Abdel Karim  
Computer Engineering Department

16

## Low-Level Readers

### ④ StringReader Class

- While it is sometimes useful to work with a character array, most programmers prefer to deal with strings.
- The StringReader class offers a substitute to the CharArrayReader, accepting a string as an input source.



Eng. Asma Abdel Karim  
Computer Engineering Department

17

# Low-Level Readers

## StringReader Class (Cont.)

- Constructors: *الString التي يتكبد عليه*
  - StringReader(String stringToBeRead)  
*Reads from the beginning of the specified string until the end.*
- Methods:  
No additional methods are added.



Eng. Asma Abdel Karim  
Computer Engineering Department

13

*بتركيبه على أي Input stream بلك تقرأ منها لنتفا تقدر تقرأ char  
مثلا bytes . ليستحسها لما*

## Low-Level Readers

*ما يكون عندي*

## InputStreamReader Class

*Crossbanding  
Reader  
for input stream*

• While readers are quite common, there is still a need for backward compatibility with older input streams.

- \*– Particularly those written by third parties for which there is no equivalent reader class. *Input stream*
- \*– For example, the System.in member variable is an InputStream instance that can read input from a user. There is no comparable reader class for this.
- \*– The solution is to connect an InputStreamReader to an InputStream instance, which will perform the necessary translation.

*from bytes into char.*



Eng. Asma Abdel Karim  
Computer Engineering Department

13

# Low-Level Readers

## InputStreamReader Class (Cont.)

- Constructors:

- `InputStreamReader(InputStream input)`

Connects an input stream to the reader.

- `InputStreamReader(InputStream input, String encoding)` throws `java.io.UnsupportedEncodingException`

Connects an input stream to the reader using the specified encoding form. If the encoding form isn't supported, an exception is thrown.

- Methods:

The `InputStreamReader` class adds the following public method:

- `String getEncoding()`

Returns the name of the character encoding used by this stream.



## InputStreamToReader Demo

```
import java.io.*;
```

```
public class InputStreamToReaderDemo {
```

```
    public static void main(String args[]){
```

```
        try{
```

```
            System.out.print ("Please enter your name : ");
```

```
            InputStream input = System.in;
```

```
            InputStreamReader reader = new InputStreamReader ( input );
```

```
            BufferedReader bufReader = new BufferedReader ( reader );
```

```
            String name = bufReader.readLine();
```

```
            System.out.println ("Pleased to meet you, " + name);
```

```
        }
```

```
        catch (IOException ioe){
```

```
            System.err.println ("I/O error : " + ioe);
```

```
        }
```

```
    }
```

```
}
```

# Filter Readers

- Filter readers, just like filter input streams provide additional functionality in the form of new methods, or process data in a different way (such as buffering).

\* Always connect to another reader.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Filter Readers

### BufferedReader Class

بجانب الـ DataInputStream  
والـ BufferedInputStream

- One of the most frustrating problems with reading data from a reader, as with an input stream, is that blocking I/O is used.
- When this happens frequently, the performance and responsiveness of software suffers. → نتائج
- An alternative is to buffer data so that reads are grouped together for better performance.
- Just as the BufferedInputStream buffers bytes of data, the BufferedReader buffers characters.

مع انه يحمي ما يوتيها فيكون  
بسي

- Also, although one would not guess it from the name, the BufferedReader is a partial substitute for the DataInputStream class.

→ It provides a readLine() method that is not deprecated.



Eng. Asma Abdel Karim  
Computer Engineering Department

# Filter Readers

## BufferedReader Class (Cont.)

- Constructors:

- **BufferedReader** (Reader reader)

*Reads data from the specified reader into a buffer.*

- **BufferedReader** (Reader reader, int bufferSize)

throws java.lang.IllegalArgumentException

*لأن أكبر من صفر لازم يكون*

*Reads data from the specified reader into a buffer, which is allocated to the specified size. The buffer size must be greater than zero.*



Eng. Asma Abdel Karim  
Computer Engineering Department

24

*قصة منه  
الreadline  
هي كالمethod  
منها الclass*

*منها منه انه هو  
Static وتغير اناسيه  
هيك*

# Filter Readers

## BufferedReader Class (Cont.)

- Methods:

- The following public method is added by BufferedReader, as a replacement for the deprecated DataInputStream.readLine() method.

- **String readLine()** throws java.io.IOException

*Reads a line of text from the underlying stream. The line is terminated by a line separator sequence, such as a carriage return/linefeed.*

- In addition, the reader overrides the markSupported() method, to indicate that it supports the mark and reset operations.



Eng. Asma Abdel Karim  
Computer Engineering Department

25

غالبا ما يستخدم ، الهدف منه لو بيدي اجمعهم اننا ال class Filter reader الخاص فيني  
ويعمل Formatting بالطريقة اللي بيدي باها.

## FilterReaders

(2)

### FilterReader Class

- Rather than perform a practical action, this class acts as a template on which other filters can be constructed.
  - If a custom filter needs to be written, the class should be extended, and methods overridden or new ones added.
- The FilterReader class has been designed so that it cannot be instantiated by making its constructor protected; the class should instead be extended.
- The FilterReader class defines no new methods, but subclasses are free to add additional methods or override existing ones.



Eng. Asma Abdel Karim  
Computer Engineering Department

26

ما بنتطرقه كثير لانه بيستخدم reset و mark

## Filter Readers

(3)

### PushBackReader Class

نفسا مبدأ  
اللي شرحناه  
قبل

- This class allows characters to be "pushed back" into the head of a reader's input queue, so that it may be read again.
- This allows programs to peek ahead at the next character and then push it back into the queue.
- Constructors:
  - **PushBackReader(Reader reader)**  
*Creates a push-back reader with a single character buffer.*
  - **PushBackReader(Reader reader, int bufferSize)** throws `java.lang.IllegalArgumentException`  
*Creates a push-back reader with a larger buffer, of the specified size. The buffer size must be greater than zero, or an exception is thrown.*



Eng. Asma Abdel Karim  
Computer Engineering Department

27

# Filter Readers

## PushBackReader Class (Cont.)

- Methods:
  - `void unread(int character)` throws `java.io.IOException`  
*Pushes the character back to the beginning of the queue. If the queue is full, an exception is thrown.*
  - `void unread(char[] charArray)` throws `java.io.IOException`  
*Pushes every character in the specified array into the queue. If full, an exception is thrown.*
  - `void unread(char[] charArray, int offset, int length)` throws `java.io.IOException`  
*Pushes a subset of the characters in the specified array into the queue, starting at the specified offset and lasting for the specified length. If full, an exception is thrown.*

# Filter Readers

## ① `LineNumberReader Class`

- The `LineNumberReader` class provides a useful line counter, which measures how many lines have been read.
- It is the reader equivalent of the `LineNumberInputStream`.
- As it extends the `BufferedReader` class, it also supports the mark/reset operations.

Counter لا update هو يعمل  
BufferReader class وورثته

# Filter Readers

## LineNumberReader Class (Cont.)

- Constructors:
  - **LineNumberReader (Reader reader)**  
*Creates a new line-number reader.*
  - **LineNumberReader (Reader reader, int size)**  
*Creates a new line-number reader and allocates a buffer of the specified size.*
- Methods:

Two new methods, to determine and to modify the line number counter, are offered.

  - **int getLineNumber()**  
*Returns the current line number.*
  - **void setLineNumber(int lineNumber)**  
*Modifies the line-number counter.*



Eng. Asma Abdel Karim  
Computer Engineering Department

30

# Filter Readers

## LineNumberReader Class (Cont.)

- Constructors:
  - **LineNumberReader (Reader reader)**  
*Creates a new line-number reader.*
  - **LineNumberReader (Reader reader int size)**  
*Creates a new line-number reader and allocates a buffer of the specified size.*
- Methods:

Two new methods, to determine and to modify the line number counter, are offered.

  - **int getLineNumber()**  
*Returns the current line number.*
  - **void setLineNumber(int lineNumber)**  
*Modifies the line-number counter.*



Eng. Asma Abdel Karim  
Computer Engineering Department

31



# The java.io.Writer Class

- Constructors:
  - There are no public constructors for this class. Instead, a writer subclass should be instantiated.
- Methods:
  - **void close()** throws java.io.IOException  
*Invokes the flush() method to send any buffered data, and then closes the writer.*
  - **void flush()** throws java.io.IOException  
*Flushes any unsent data, sending it immediately. A buffered writer might not yet have enough data to send, and may be storing it for later.*



Eng. Asma Abdel Karim  
Computer Engineering Department

موضوعات ما يتعلق  
البرنامج

32

# The java.io.Writer Class

low level writing.

- Methods:
  - **void write(int character)** throws java.io.IOException  
*Writes the specified character.*
  - **void write(char[] charArray)** throws java.io.IOException  
*Writes the entire contents of the specified character array and writes it.*
  - **void write(char[] charArray, int offset, int length)** throws java.io.IOException  
*\* Reads a subset of the character array, starting at the specified offset and lasting for the specified length, and writes it.*
  - **void write(String string)** throws java.io.IOException  
*Writes the specified string.*
  - **void write(String string, int offset, int length)** throws java.io.IOException  
*Writes a subset of the string, starting from the specified offset and lasting for the specified length.*

تقدير على low level  
تكتب string

Eng. Asma Abdel Karim  
Computer Engineering Department

33

بشبه المصفوفة Byte Array Output Stream  
 ، وظيفتهم يكتبوا s Array  
 هم من كالم عملوا ال Array  
 التي يكتبوا عليها

# Low-Level Writers

## The CharArrayWriter Class

فغشان هيك  
 ال constructor  
 ما ياتنا  
 Array of bytes  
 كبره داخذ  
 ال buffer size  
 ولو بيدي  
 ال string  
 ال writer

The CharArrayWriter maintains an internal buffer that is added to each time a write request is made, and may be converted to a character array.

Constructors:

- **CharArrayWriter()**  
Creates a character array writer that can be converted to a character array.
- **CharArrayWriter(int bufferSize)** throws java.lang.IllegalArgumentException  
Creates a character array writer using the specified initial buffer size (which must not be negative).

Methods:

- **char[] toCharArray()**  
Returns a character array, containing all characters written thus far.
- **String toString()**  
Returns a string containing all characters written thus far.

Byte Array Output Stream بال اسمها "to Byte Array"



Eng. Asma Abdel Karim  
 Computer Engineering Department

# Low-Level Writers

## The FileWriter Class

- The FileWriter class extends the OutputStreamWriter class, and provides a convenient way to write characters to a local file.
- This class is equivalent to the FileOutputStream class discussed earlier.

Constructors:

- **FileWriter (File file)** throws java.io.IOException  
Creates a writer connected to the resource represented by the specified file object if not prevented by security permissions.
- **FileWriter (FileDescriptor descriptor)** throws java.io.IOException  
Creates a writer connected to the specified descriptor handle, if allowable.
- **FileWriter (String filename)** throws java.io.IOException  
Writes to the specified file location, creating a file if one does not already exist and overwriting an existing one. If not permitted by security access restrictions, an exception will be thrown.
- **FileWriter (String filename, boolean appendFlag)** throws java.io.IOException  
Writes to the specified file location. If the appendFlag is set to "true," the file will be opened in append mode and data will be written to the end of the file.

"هذا يستخدم"

by default = false

\* لو ال File موجود عندك طبعاً تكتب عليه وفيه data كان ، ما يرمه بيلش  
 يكتب من الاول ويشيلها ، لو ما بيدي اشي كان مكتوب  
 على ال File يستخدم هاي ال Method وبيدي ال appendFlag تكون true  
 فبس بيلش يكتب فما يكتب فوق الموجود لا يضاف اليه.



Eng. Asma Abdel Karim  
 Computer Engineering Department

Threads

## Low-Level Writers

### ③ The PipedWriter Class

لنستوفهم  
بغير

- The purpose of the PipedWriter class is to write data that will be read by a PipedReader.
- These two classes are reader/writer equivalents of the PipedInputStream and PipedOutputStream classes, but may not be interchanged.
- Constructors:

- **PipedWriter()**

Creates an unconnected pipe writer.

ال reader التي تتربط عليه

اللي جاي من thread ثاني .

- **PipedWriter(PipedReader reader)** throws java.io.IOException

Creates a piped writer connected to the specified reader. The reader may later read any data written to this writer.

و بتشكروهم مع بعض

- Methods:

- **void connect (PipedReader reader)** throws java.io.IOException

Attempts to connect to the specified pipe, so that any data written may be read by the reader. If the pipe is already connected to another pipe, an IOException will be thrown.

\* عموما في PipedWriter بتربط PipedReader أو PipedWriter

بتربط PipedReaders ، لازم PipedReaders بتربط

PipedWriter أو العكس. 36



Eng. Asma Abdel Karim  
Computer Engineering Department

## Low-Level Writers

### ④ The StringWriter Class

\* ال String فيها بيانه زي ال Array كل اضافة بتعمل

String جديد وورد يضيف عليه مكان هيك enternally

- Judging by its name, you might expect that this class allowed for writing to a string.
- A string is of fixed length and is immutable (the contents of a string may not be modified).
- Writing to a string is accomplished by using a StringBuffer.

يكون في StringBuffer

بخل يتكتب عليه من ال String الاساسي

- The StringBuffer class is similar to a string, but may be modified. When the modifications are complete, the StringBuffer can be converted back to a string.

لعد ما انت تطلب ال String بوديك

- This is how the StringWriter class works.

ياه ال String الاملي و بتجعلك ياه

- It maintains a string buffer, and provides a method to access the buffer contents or to convert to a string.



Eng. Asma Abdel Karim  
Computer Engineering Department

37

# Low-Level Writers

## The StringWriter Class

- Constructors:
  - **StringWriter()**  
Creates a new string writer, using the default-sized buffer.
  - **StringWriter(int startingSize)**  
Creates a new string writer and allocates a StringBuffer the specified size.
- Methods:
  - **StringBuffer getBuffer()** → *ترجع ال Buffer التي كتبت عليه قبل ما ينقله لـ String*  
Returns the buffer used to store data sent to the writer.
  - **String toString()** → *ترجع ال String النهائي*  
Converts the internal buffer into a string.



Eng. Asma Abdel Karim  
Computer Engineering Department

*output stream writer all to corresponding*

## Low-Level Writers

## **The OutputStreamWriter Class**

- While there are many writer classes equivalent to output stream classes in the Java API, there is still a need to maintain compatibility with older output stream classes.
- \* – As most of the networking API and some third-party libraries provide only stream interfaces.
- The OutputStreamWriter class handles translation between a Writer and OutputStream, allowing new writer classes to interact with older output streams.

*char to byte & byte to output stream*



Eng. Asma Abdel Karim  
Computer Engineering Department

# Low-Level Writers

## The OutputStreamWriter Class

- Constructors:

- `OutputStreamWriter(OutputStream output)`

Creates a writer that will translate between characters and bytes, using the default character encoding. → Unicode

- `OutputStreamWriter(OutputStream output, String encoding)` throws `java.io.UnsupportedEncodingException`

Creates a writer that translates between characters and bytes, using the specified character encoding. If the specified encoding form is not supported, an exception is thrown.

لو بي  
أغير طريقة  
الencoding

- Methods:

- `String getEncoding()`

Returns the character encoding used by the writer.



Eng. Asma Abdel Karim  
Computer Engineering Department

40

## OutputStreamToWriter Demo

```
import java.io.*;
public class OutputStreamToWriterDemo
{
    public static void main(String args[])
    {
        try
        {
            //Get the output stream representing standard output
            OutputStream output = System.out; → Additional Step
            // Create an OutputStreamWriter
            OutputStreamWriter writer = new OutputStreamWriter (output);
            // Write to standard output using a writer
            writer.write ("Hello world");
            // Flush and writer, to ensure it is written
            writer.flush(); writer.close();
        }
        catch (IOException ioe)
        {
            System.err.println ("I/O error : " + ioe);
        }
    }
}
```

ممكن تحتاج  
flush  
حسب  
System.out



Eng. Asma Abdel Karim  
Computer Engineering Department

41

نفس مبدأ القارئ  
لوبيدي ان على buffered output لا تنوي

1

## Filter Writers

### The BufferedWriter Class

• Used to improve system performance by buffering write request together.

• Constructors:

\* - **BufferedWriter(Writer writer)**

Creates a buffered writer, connected to the specified writer. Write requests will be buffered, to improve efficiency. To send all queued data, the flush() method should be invoked.

← مهمة اناديها

\* - **BufferedWriter(Writer writer, int bufferSize)** throws java.lang.IllegalArgumentException

Creates a buffered writer, with a buffer of the specified size. The size must be greater than or equal to 1.



like Filter Reader ←

2

## Filter Writers

### The FilterWriter Class

• Developers creating custom filter classes should extend this class, rather than extending the java.io.Writer class.

• It provides no additional functionality, but may be used as a template on which filters can be constructed.

• Constructors

ما يقبل استخدام Create new object

\* - **protected FilterWriter(Writer writer)** بال class تاسه

• Methods:

- The FilterWriter class defines no new methods, but subclasses are free to add additional methods or override existing ones.



## 3 Filter Writers

### The PrintWriter Class

بيكتال data  
اللي بيديها شكل text

- PrintWriter is the sister class of PrintStream, and provides the same methods for writing datatypes as text.
- Like PrintStream, none of the methods may throw an IOException—rather, the error state is determined by invoking the checkError() method, which returns a boolean value.
- Constructors:
  - PrintWriter(Writer writer)  
Creates a print writer, writing to the specified writer.
  - PrintWriter(Writer writer, boolean flushFlag) → by default = False  
Creates a print writer, the output of which may or may not be automatically flushed whenever a println() method or a line separator is sent, based on the state of the specified boolean flag. A value of "true" will flush when a println method is executed.
- Methods:
  - The PrintWriter class implements new methods to match the signatures of the PrintStream class.



تفسر ال object كامل وتكتبه على Stream كسك bytes

## Object Persistence & Object Serialization

عن طريق ال low level

- Data that can be read or written ranges from individual bytes to primitive datatypes and strings. → عن طريق ال filters
- But what if you wanted to read and write an entire object, composed of a series of member variables?
- To do this would require that each field of the object be written individually; then at a later time, each field would be read back and assigned to an object.
- \* This is a complicated process. → لكن معقد جدا بالمرتبطة
- \* The solution is to use object persistence. → الحل = object serialization

زوي  
ما تظن  
عاب



فكرة إنه ال System يخل بده قدرة  
على البقاء حتى لو انتهى  
تنفيذ البرنامج.

## What is Object Persistence

- حتى لو خلا ال App تنفيذ يظل العنصر مخزن بمكان ونقدر أقرأه مرة ثانية
- Object persistence is the ability of an object to persist over time (and, if moved to a different computer or JVM, over space). → بقدر أنقل ال object من جvm إلى آخر
  - Most objects in a Java Virtual Machine are fairly short-lived.
  - When there are no references to an object, the memory space allocated to it is reclaimed by the automatic garbage collector thread.
    - 1
  - If an object is frequently used, and does not lose references to it, it will still die at some point in time the JVM will terminate eventually and the object will be destroyed.
    - 2
  - Object persistence allows an object to outlive the JVM that hosts it.

ال object يعيش أكثر من ال JVM التي أنشأته بإني  
أحفظه بـ File مثلاً.



Eng. Asma Abdel Karim  
Computer Engineering Department

46

هو المسؤول بيف عالو ال Variables المختلفة شو ما كانوا ويكتبون بشكل

أقربها في قبل  
اللي فوقها.

## What is Object Serialization

Array of bytes

- Object serialization controls how the data that comprises an object's state information (the individual member variables, whether public, private, or protected) is written as a sequence of bytes.

ال objects

بقدر الكتبتهم

بأي مكان

عن طريق ال

Serialization

The serialized object might be sent over the network or saved to a disk so that it can be accessed at some point in the future.

This allows objects to move from one JVM to another, whether located on the same machine or a remote one.

فكرة ال object إني أقدر أأخذ ال object بـ File وأقرأه متى ما بدني

أو إني أنقل ال object من مكان إلى آخر هي

اللي بتبطيني فكرة ال object  
Persistence



Eng. Asma Abdel Karim  
Computer Engineering Department

47

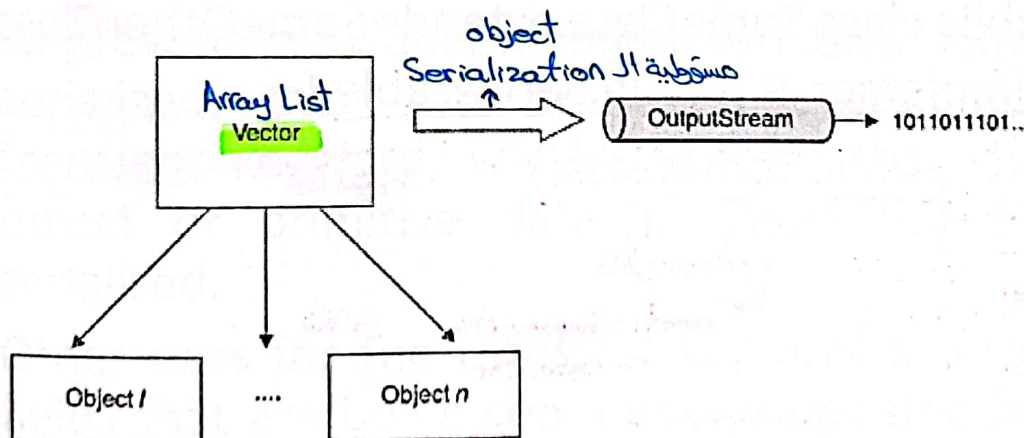


# What is Object Serialization (Cont.)

- Serialization works by examining the variables of an object and writing primitive datatypes like numbers and characters to a byte stream. → هويج تعمل هاد عناء
- If an object contains an object as a member variable (as opposed to a primitive datatype), the object member variable must be serialized as well. → لولقت في علاقة بين ال objects  
Serialization  
كلوم برينه
- This must be done recursively, so that if an object has a reference to an object, which has a reference to another object (and so on), they are all saved together.
- The set of all objects referenced is called a graph of objects, and object serialization converts entire graphs to byte form. \*



## Graphs of Objects



الاشياء التي لازم تكون  
 في ال object  
 كمان نقدر نعمله  
 object serialization: ①, ②

# How Serialization Works?

- Support for serialization was introduced in JDK1.1. ①
- Any object that implements the java.io.Serializable interface may be serialized with only a few lines of code (along with any other object referenced by a serialized object).
- The interface serves only as an indication that the developer endorses serialization—no methods need to be implemented to support serialization. → Method جواته لتعملها
- Implementing the java.io.Serializable interface requires no additional effort on the part of developers, other than:
  - ① = Adding the appropriate "implements" statement during the class declaration and,
  - ② = Declaring a no-argument constructor (also referred to as the default constructor).
- The constructor is required so that the class maybe instantiated later by the JVM, and then deserialized by assigning new values to member variables.

لعمل العملية الحاسوبية بتحويل bytes  
 ل object التي كان عليه  
 Eng. Asma Abdel Karim  
 Computer Engineering Department

## Example

```

    P ال ←
    بتكون Small
    Public class SomeClass extends SomeOtherClass
    implements java.io.Serializable {
    public SomeClass()
    {
    }
    .....
    }
    
```

انما هال object  
 able to be  
 serialization

اهل شرط  
 default no Arguments  
 constructor

يكون في

Eng. Asma Abdel Karim  
 Computer Engineering Department



# Serialization Issues

- There are some legitimate reasons, too, for not supporting serialization.
    - \* - For example, if an object contained very sensitive information, it might be unwise to serialize it and save it to disk or send it over a network.
    - \* - Developers should be aware that no special care is taken to protect the contents of a serialized object from scrutiny or modification, and that any class in any JVM may choose to deserialize an object at a later time.
- ما في أي نوع من ال Security بحال  
Serialize ال

## Serialization Issues (Cont.)

- To prevent individual member variables being serialized, they can be marked with the transient keyword, which indicates that the object or primitive datatype must not be serialized.
  - Other uses for the *transient* keyword are for fields that are being continuously updated by some means, such as a timer, and hence do not make sense to serialize.
- يستخدموا ال data field اللي ما بي يتغيروا Serialize  
فيم يتغير  
لنك يفضل  
ما تفعلوا Serialize برضك.

# Example

Public class UserAccount implements  
java.io.Serializable {  
protected String username;  
protected transient String password;

```
public UserAccount( )  
{  
    ...  
}
```

↓  
Pass ما يظهر ال  
default يظهر ال  
تبع الةة : مثلا لو  
String يطالبك : null



Eng. Asma Abdel Karim  
Computer Engineering Department

## Reading and Writing Objects to Streams

- The main point of serialization is to write an object out to a stream and to read it back.
- This is accomplished by using the java.io.ObjectOutputStream and java.io.ObjectInputStream classes, which can write serializable objects out to an output stream and read them back from an input stream.

تركبوا فوق  
level  
Stream أو بالآخر  
مربوطين ب  
level  
Filter  
Streams  
اقرأ  
object  
deserialization

الكتب ال  
Serialization



Eng. Asma Abdel Karim  
Computer Engineering Department

# The ObjectInputStream Class

- The ObjectInputStream class is used to read a serialized object from a byte stream, to allow an object to be reconstituted back to its original form, [providing the object's class can be loaded by the JVM's class loader.] *لازم يكون ال class موجود بالنطاق وقادر على تحميله*
- The ObjectInputStream class implements the ObjectInput interface, which extends the DataInput interface. *لذا Data Input + Methods to read objects*
- This means that the ObjectInputStream class provides many methods with the same signature as DataInputStream, in addition to extra methods responsible for reading objects.



## The ObjectInputStream Class (Cont.)

- Constructors: *ملاحظة*
  - protected ObjectInputStream() throws `java.io.IOException` `java.lang.SecurityException`  
Provides a default constructor for `ObjectInputStream` subclasses.
  - ObjectInputStream(InputStream input) throws `java.io.IOException` *ملاحظة*  
Creates an object input stream connected to the specified input stream, which is capable of restoring serialized objects.



## The ObjectOutputStream Class (Cont.)

- Methods:

Many of the methods of *ObjectInputStream* were covered in the discussion of the *DataInputStream* class.

*ObjectInputStream* can read primitive datatypes just like the *DataInputStream* class.

- public final Object **readObject()** throws java.io.OptionalDataException, java.io.IOException, java.lang.ClassNotFoundException

→ بترجولي object بشكل عام  
لو بيدي النوع الفعلي ليستخدم  
Downcasting

\* Reads a serialized object from the stream and reconstructs it to its original state. If the object contains references to other objects, these objects are also reconstructed. If an object cannot be read, the application will be notified by the method throwing an exception. [An Object instance is returned. If required, this object can be cast to a specific class type before it is used.]



Eng. Asma Abdel Karim  
Computer Engineering Department

58

← أهم لأنه فيراني بقدر أكتب كذا اللي قبله + يكتب محتوي

## The ObjectOutputStream Class

عالية الكتابة

- The *ObjectOutputStream* class serializes an object to a byte stream, for the purpose of object persistence.
- It may be connected to any existing output stream, such as a file or a networking stream, for transmission over the Internet.
- Objects written to an *ObjectOutputStream* have all their member variables (such as primitive data types and objects) written.
- If the object contains references to other objects, they too will be written, so an *ObjectOutputStream* can write entire object graphs.
- A sequence of objects can be written or wrapped in a collection (such as an array or a vector) whose entire contents could be serialized with one call to the *ObjectOutputStream.writeObject* method.

or Array List



Eng. Asma Abdel Karim  
Computer Engineering Department

59

## The ObjectOutputStream Class (Cont.)

- Constructors:

- `protected ObjectOutputStream ()` throws `java.io.IOException` `java.lang.SecurityException`

*Default constructor, provided for the benefit of subclasses of the `ObjectOutputStream`.*

- `ObjectOutputStream (OutputStream output)` throws `java.io.IOException`

*Creates an object output stream capable of serializing objects to the specified output stream.*



## The ObjectOutputStream Class (Cont.)

- Methods:

- The `ObjectOutputStream` class also provides method implementations for the `DataOutput` interface.

- `void writeObject (Object object)` throws `java.io.IOException`, `java.io.InvalidClassException`, `java.io.NotSerializableException`

*\* Writes the specified object to the output stream, through object serialization. All variables that are not marked as transient or static will be written, providing the specified class is an instance of the `java.io.Serializable` interface.*



# References

Chapter 4 of *Java™ Network Programming and Distributed Computing*, David Reilly and Michael Reilly.



Eng. Asma Abdel Karim  
Computer Engineering Department



# Networks and Internet Programming

## User Datagram Protocol

↓  
Transport Layer  
Protocol



Eng. Asma Abdel Karim  
Computer Engineering Department

نكتب برامج والشبكات  
طريقا ل UDP

1

## Outline

- Overview.
- DatagramPacket Class.
- DatagramSocket Class.
- Listening for UDP Packets.
- Sending UDP Packets.
- Additional Information on UDP.



Eng. Asma Abdel Karim  
Computer Engineering Department

2

\* TCP Socket : بس تفتحها مجاناً بيبي معها  
output Stream و InputStream

\* UDP Socket : يستخدم طريقة مختلفة جداً تسمى Packet  
Streams بس يكتب وبقدر عال Packet

فيوم

## Overview

- The User Datagram Protocol (UDP) is a commonly used transport protocol employed by many types of applications. → connection oriented كس
- UDP is a connectionless transport protocol, meaning that it doesn't guarantee either packet delivery or that packets arrive in sequential order.
- Rather than reading from, and writing to, an ordered sequence of bytes using I/O streams, bytes of data are grouped together in discrete packets, which are sent over the network.

← لما بنحكي عال UDP بالباية ما يكون فيه تشبيه بين الطرفين انهم جيسروا يتكلموا مع بعض اوهيك بلعيب فجأة قادر التواصل.

ال UDP ما يتحلل قاعى المشاكل



Eng. Asma Abdel Karim  
Computer Engineering Department

3

## Overview

- The packets may travel along different paths, as selected by the various network routers that distribute traffic flow, depending on factors such as network congestion, priority of routes, and cost of transmission.
- \* - This means that a packet can arrive out of sequence, if it encounters a faster route than the previous packet (or if the previous packet encounters some other form of delay).
- \* - No two packets are guaranteed the same route, and if a particular route is heavily congested, the packet may be discarded entirely. Each packet has a time-to-live (TTL) counter, which is updated when the packet is routed along to the next point in the network. When the timer expires, it will be discarded, and the recipient of the packet will not be notified.
- \* - If a packet does arrive, however, it will always arrive intact. Packets that are corrupt or only partially delivered are discarded.

← مش ضروري توصل بشكل صحيح وانما.

\* هاي مش اخطاء ال UDP هاي امور الشبكة.



Eng. Asma Abdel Karim  
Computer Engineering Department

4

**فوائد حتمًا لوضع اشياء من الـ packet**

## Advantages of UDP

بعضها جالبي قبل  
بأنها تكررت بشكل  
كبير

- UDP communication can be more efficient than guaranteed-delivery data streams. If the amount of data is small and the data is sent frequently.
- Unlike TCP streams, which establish a connection, UDP causes fewer overheads.
  - \* - If the amount of data being sent is small and the data is sent infrequently, the overhead of establishing a connection might not be worth it.
  - \* - If data is being sent from a large number of machines to one central one, in which case the sum total of all these connections might cause significant overload.
- Real-time applications that demand up-to-the-second or better performance may be candidates for UDP, as there are fewer delays due to the error checking and flow control of TCP.
- UDP sockets can receive data from more than one host machine. If several machines must be communicated with, then UDP may be more convenient than other mechanisms such as TCP.
- Some network protocols specify UDP as the transport mechanism, requiring its use.

UDP أقل overhead من الـ TCP

هون مجبور تستخدم UDP لإنهاء الـ App  
Protocol يكون منفي بشي يستخدم . مثل DNS يستخدم .  
UDP

Eng. Asma Abdel Karim  
Computer Engineering Department

أفضل استخدام الـ UDP لأنه مثلاً الـ Video أو صوتية  
Frame على صريح مارج ياتش هالتأثير فعادي + الـ redundancy

بنتكون موجودة على مستوى الـ App نفسه بدون  
ما احتاج أسأل الطرف الثاني وأطول الوقت.

## java.net.DatagramPacket Class

الوحدة التي حنحضرها للبرمال أو الاستقبال

- The DatagramPacket class represents a data packet intended for transmission using the User Datagram Protocol.
- Packets are containers for a small sequence of bytes, and include addressing information such as an IP address and a port.

\* فلسفة التشغيل بال  
UDP و بجزء الـ Packet  
لديني يا إما يرسلوا  
أو يستقبلوا عن  
طريق الـ Socket

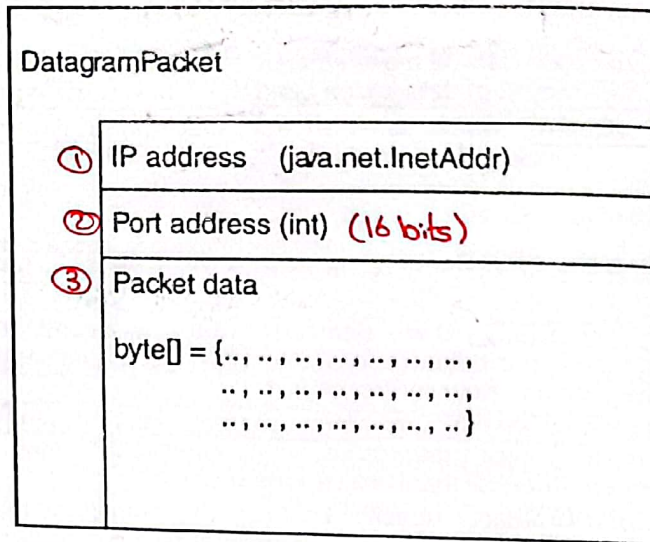
أي Socket بفتحها  
لازم الـ Port  
Number  
لغير الـ data  
رايحة لأي Socket  
أو جاري من أي Socket

بخط فيها الـ data وبعدها الـ port  
يرسلوا أو يتطلب يخط جزء الأشياء التي  
وألمة لو بدو يستقبل

Eng. Asma Abdel Karim  
Computer Engineering Department

## الـ Datagram Packet الـ DataField

# java.net.DatagramPacket Class



Eng. Asma Abdel Karim  
Computer Engineering Department

\* للتواصل بين طرفين نحتاج اكيو 7

ولكننا نحتاج عنا واحد بس ، ولاننا بنخزن فيهم معلومات الطرف الثاني . لان معلومات الطرف الاول بتكون معروفة بال Socket .

# java.net.DatagramPacket Class

- The meaning of the <sup>Addressing Info (IP/Port)</sup> ~~data~~ stored in a **DatagramPacket** is determined by its context.
- When a <sup>مستقبلة</sup> **DatagramPacket** has been read from a UDP socket, the IP address of the packet represents the address of the sender (likewise with the port number).
- However, when a <sup>مُرسلَة</sup> **DatagramPacket** is used to send a UDP packet, the IP address stored in **DatagramPacket** represents the address of the recipient (likewise with the port number).

← لما استقبل  
datagrampacket  
يكون متزنا بال  
datafield معلومات  
الـ Sender  
والعكس .



Eng. Asma Abdel Karim  
Computer Engineering Department

# Creating a DatagramPacket

There are two reasons to create a new **DatagramPacket**:

1. To send data to a remote machine using UDP.
2. To receive data sent by a remote machine using UDP.

يعني لو استخدمت اول constructor للرسال مارج يعني انه  
 error والمخس بس الافضل انه استخدم ال constructor  
 زي ما مشروح تحت



Eng. Asma Abdel Karim  
Computer Engineering Department

\* في حالة ال Sending انت اللي بتعطي ال datagram packet بينما حالة ال receiving  
 انت بتعمل DatagramPacket بس جواها فاضي مش انت بتعطيها بتعطيها لل UDP  
 وهو بيملك معلومات الطرف الثاني

# Creating a DatagramPacket

مش مخصصين بس لتلقي العملية  
 من ال غالب هيك  
 حتم استخدام  
 كل ال constructors

Constructors:

- The choice of which DatagramPacket constructor to use is determined by its intended purpose.
- Either constructor requires the specification of a byte array, which will be used to store the UDP packet contents, and the length of the data packet.
- ⊖ To create a DatagramPacket for receiving incoming UDP packets, the following constructor should be used:

1. **DatagramPacket(byte[] buffer, int length).**

For example: `DatagramPacket packet = new DatagramPacket(new byte[256], 256);`  
 ↳ Array of data

- To send a DatagramPacket to a remote machine, it is preferable to use the following constructor:

2. **DatagramPacket(byte[] buffer, int length, InetAddress dest\_addr, int dest\_port).**

For example: `InetAddress addr = InetAddress.getByName("192.168.0.1");`  
`DatagramPacket packet = new DatagramPacket ( new byte[128],128, addr, 2000);`  
 (IP address) object      Port Num.

بال Socket لقدام  
 بتوضح الامور  
 بال Receiving عاينة  
 بخطهم نفس الحجم  
 وبين يعطيو هو المعلومات  
 لو كان طول ال data اقل  
 هو بيمل  
 عليه

لانه ممكن اعمل  
 Array كبيرة تشمل  
 الكل وجواها يعطي  
 ال data لانه ممكن  
 ال Array تكون اكبر  
 من ال data

قبل ما اعله لازم اعطي ال data وفي طرق مختلفة  
 للتعبئة وفي طريقة  
 بعد الإنشاء باستخدام ال Setter



Eng. Asma Abdel Karim  
Computer Engineering Department

## Creating a DatagramPacket

### • Methods:

– The **DatagramPacket** class provides some important methods that allow the remote address, remote port, data (as a byte array), and length of the packet to be retrieved.

– As of JDK1.1, there are also methods to modify these, via a corresponding set method. This means that a received packet can be reused.

• For example, a packet's contents can be replaced and then sent back to the sender. This saves having to reset addressing information—the address and port of the packet are already set to those of the sender.

← استرجاع  
← لو بي أرد عالي و طيفي بيأما لعل Packet جديدة  
← استفيد منها بالبرايه والردود  
← بعيد الاستخدام نفس اللي و طيفي



Eng. Asma Abdel Karim  
Computer Engineering Department

11

و بعمل Set لاد احدثه الجديده وبعمل Send

## Creating a DatagramPacket

### • Methods:

– **InetAddress getAddress()**

Returns the IP address from which a DatagramPacket was sent, or (if the packet is going to be sent to a remote machine), the destination IP address.

→ ناع الطرف الثاني  
– **byte[] getData()**

Returns the contents of the DatagramPacket, represented as an array of bytes.

– **int getLength()**

Returns the length of the data stored in a DatagramPacket. This can be less than the actual size of the data buffer.

– **int getPort()**

Returns the port number from which a DatagramPacket was sent, or (if the packet is going to be sent to a remote machine), the destination port number.



Eng. Asma Abdel Karim  
Computer Engineering Department

12

## Creating a DatagramPacket

### Methods:

- void **setAddress(InetAddress addr)**

Assigns a new destination address to a DatagramPacket.

- void **setData(byte[] buffer)** → لازم يكون كافي لحجم ال data

Assigns a new data buffer to the DatagramPacket. Remember to make the buffer long enough, to prevent data loss.

- void **setLength(int length)**

Assigns a new length to the DatagramPacket. Remember that the length must be less than or equal to the maximum size of the data buffer, or an `IllegalArgumentException` will be thrown. When sending a smaller amount of data, you can adjust the length to fit—you do not need to resize the data buffer.

- void **setPort(int port)**

Assigns a new destination port to a DatagramPacket.



الاستخدام

← أهم وموجود دائما ويعتبر جسر بين ال App وال transport layer

### java.net.DatagramSocket Class

- The **DatagramSocket** class provides access to a UDP socket, which allows UDP packets to be sent and received.
- A **DatagramPacket** is used to represent a UDP packet, and must be created prior to receiving any packets.
- \* The same **DatagramSocket** can be used to receive packets as well as to send them.
- Read operations are **blocking**, meaning that the application will continue to wait until a packet arrives.
  - Since UDP packets do not guarantee delivery, this can cause an application to stall if the sender does not resubmit packets.
  - You can use multiple threads of execution, or as of JDK1.1, you can use nonblocking I/O to avoid this problem.



بال كده تفتح Socket عن طريقها برسلك ويستقبل datagrampacket بنحضر ال Packet قبل ال Socket \*

\* معلومات ال flexible addressing بتكون على مستوى ال Socket مثل ال Socket

كشافي بقرا من الشبكة فما في ضمانه لوصول ال data كذلك ال كده ما فيها ضمانات للوصول فممكن يدخل block للآب .

بنتاولها باستخدام حلول مختلفه مثل ال timer وال threads

\* أي Socket بفتحها لازم ال Port Num خلاصا زيها مختلف الالوجوز محدوز الال اليبصرها ال Transport Layer \*

## Creating a DatagramSocket

- A **DatagramSocket** can be used to both send and receive packets.
- Each **DatagramSocket** binds to a port on the local machine, which is used for addressing packets.
- The port number need not match the port number of the remote machine, but if the application is a UDP (server) it will usually choose a specific port number.
- ② If the **DatagramSocket** is intended to be a client and doesn't need to bind to a specific port number, a blank constructor can be specified.

هو شرط لو  
2 devices  
لدهم يحكو مع  
بعض انه يكون  
ال Port Number  
ال Matching

يعني ما يباخذ اشئ  
لانه هو اللي بيلش يحكي وما في داعي  
الطرف الثاني

يكون عارف ال Port Num تاه بس هو لازم يكون عارف ال Port Num تاع الطرف الثاني  
ولمجرد ما يكون جهت للطرف الثاني بحس الطرف الثاني لغير يرد عليه.

## Creating a DatagramSocket (Cont.)

- To create a client DatagramSocket, the following constructor is used:
- ① **DatagramSocket() throws java.net.SocketException**
- To create a server Datagram Socket, the following constructor is used, which takes as a parameter the port to which the UDP service will be bound:
- ② **DatagramSocket(int port) throws java.net.SocketException**

Port Num عادي  
Port Num عادي  
Port Num عادي  
بناخد Randomly  
بكون

Super Class  
لerror  
أمنى.

نفس ملاحظة ال Slide ، انه مش  
اجباري تستخدم هيك زي ما انكتب واكن هيك الاصل



\* بـ ١ و ٢ لو كان في IP Address واحد عاليجان يكون هو اللي يتكوي عليه ال Socket ولو كان في أكثر من IP Address بيأخذ واحد منهم random وبيشيك عليه، لو بدى أحدد أنا أي IP Address بيستخدم بيستخدم ٣

## Creating a DatagramSocket (Cont.)

- Although rarely used, there is a third constructor for **DatagramSocket**, introduced in JDK1.1.
- If a machine is known by several IP addresses, you can specify the IP address and port to which a UDP service should be bound.
- It takes as parameters the port to which the UDP service will be bound, as well as the InetAddress of the service.
- This constructor is:

③ **DatagramSocket (int port, InetAddress addr) throws java.net.SocketException**



Eng. Asma Abdel Karim  
Computer Engineering Department

17

## \* Using a DatagramSocket

- **DatagramSocket** is used to receive incoming UDP packets and to send outgoing UDP packets.
- It provides methods to:
  - ① - Send and receive packets, <sup>أهم ٢</sup>
  - ② - Specify a timeout value when nonblocking I/O is being used, <sup>لما ما يعمل blocking للنوعية</sup>
  - ③ - Inspect and modify maximum UDP packet sizes, and
  - ④ - Close the socket.

لأنه ال Socket يتحرق  
resources فينسكفها  
بها نخلص



Eng. Asma Abdel Karim  
Computer Engineering Department



## Using a DatagramSocket (Cont.)

Socket ←

- `int getSoTimeout()` throws `java.net.SocketException`

بذلك اني السيت ال time و ما واط  
اشي

Returns the value of the timeout socket option. This value is used to determine the number of milliseconds a read operation will block before throwing a `java.io.InterruptedIOException`. By default, this value will be zero, indicating that blocking I/O will be used.

- `void receive(DatagramPacket packet)` throws `java.io.IOException`

Reads a UDP packet and stores the contents in the specified packet. The address and port fields of the packet will be overwritten with the sender address and port fields, and the length field of the packet will contain the length of the original packet, which can be less than the size of the packet's byte-array. If a timeout value hasn't been specified by using `DatagramSocket.setSoTimeout(int duration)`, this method will block indefinitely. If a timeout value has been specified, a `java.io.InterruptedIOException` will be thrown if the time is exceeded.

ال \* receive  
وال Send هم  
المومينا التي يعملها  
عمليات الإرسال  
والاستقبال

← ياخذ محتويات ال packet و يخزنهم في ال datagram packet



Eng. Asma Abdel Karim  
Computer Engineering Department

object → App مستوى ال → 21 مش نفس ال  
UDP packet

## Using a DatagramSocket (Cont.)

→

كس ال receive

- `void send(DatagramPacket packet)` throws `java.io.IOException`

Sends a UDP packet, represented by the specified packet parameter.

- `void setReceiveBufferSize(int length)` throws `java.net.SocketException`

Sets the maximum buffer size used for incoming UDP packets. Whether the specified length will be adhered to is dependent on the operating system.

- `void setSendBufferSize(int length)` throws `java.net.SocketException`

Sets the maximum buffer size used for outgoing UDP packets. Whether the specified length will be adhered to is dependent on the operating system.

- `void setSoTimeout(int duration)` throws `java.net.SocketException`

Sets the value of the timeout socket option. This value is the number of milliseconds a read operation will block before throwing a `java.io.InterruptedIOException`.

لو ال length التي  
كلية يا مش متوافق  
مع ال OS مش شرج  
انه يعطيك ال  
length التي طلبته



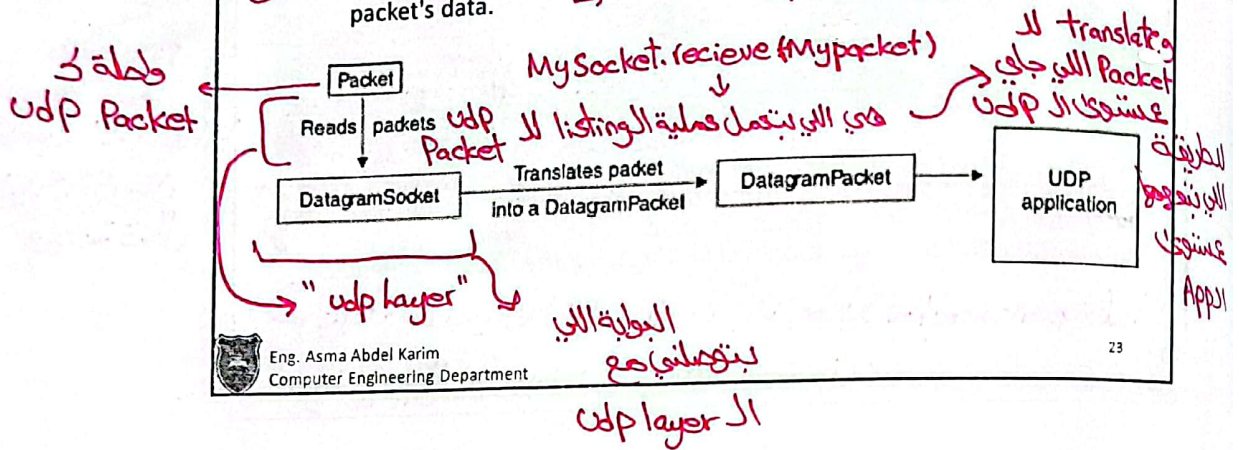
Eng. Asma Abdel Karim  
Computer Engineering Department

بعد time معين او خلصت ما واطت  
ال data تمشي و بكدها ينزل blocking 22

(بتخير عليه ال receive ال كونه ب time)

# Listening for UDP Packets

- Before an application can read UDP packets sent to it by remote machines, it must:
  - \* - Bind a socket to a local UDP port using `DatagramSocket`, and
  - \* - Create a `DatagramPacket` that will act as a container for the UDP packet's data.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Listening for UDP Packets (Cont.)

- When an application wishes to read UDP packets, it calls the `DatagramSocket.receive` method, which copies a UDP packet into the specified `DatagramPacket`. The contents of the `DatagramPacket` are processed, and the process is repeated as needed.

```

DatagramPacket packet = new DatagramPacket(new byte[256], 256);
DatagramSocket socket = new DatagramSocket(2000);
boolean finished = false;
while (!finished)
{
    socket.receive(packet); // blocking
    // process the packet
}
socket.close();
    
```

Packet جوا الـ Array of bytes  
Stream بركب  
data لآقى الـ  
Format لآقى  
علي  
للإستقبال

length الـ  
المبني  
(max الـ)

حسب أنا بعينها  
لتنصيح معلومات الـ Packet الذي  
الاستقبلت فيها عن طريق الـ receive

Eng. Asma Abdel Karim  
Computer Engineering Department

لأنه يكون في توافق ، لو كتبت  
read int write int باستخدام  
(consistency) وهكذا

## Listening for UDP Packets (Cont.)

- When processing the packet, the application must work directly with an array of bytes.
- If, however, your application is better suited to reading text, you can use classes from the Java I/O package to convert between a byte array and another type of stream or reader.
  - By hooking a `ByteArrayInputStream` to the contents of a datagram and then to another type of `InputStream` or an `InputStreamReader`, you can access the contents of UDP packets relatively easily.



```

ByteArrayInputStream bin = new ByteArrayInputStream(
packet.getData());
DataInputStream din = new DataInputStream(bin);
// Read the contents of the UDP packet
    
```

كشافة من جهة الـ Sender تركيب DataOutputStream

ضروري اول وحدة



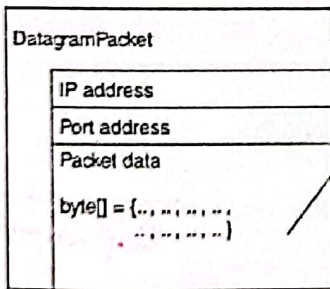
Eng. Asma Abdel Karim  
Computer Engineering Department

\* لازم يكون فيه تساقبين عملية القراءة والكتابة

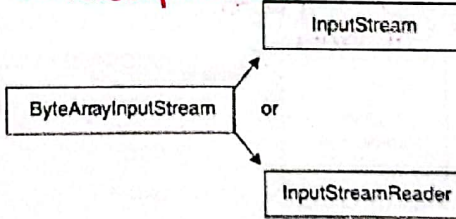
لو تركيب writeInt و ReadInt استخدم وهكذا

لانه الطرفين بنكتبوا بنفس الـ teams فيعرفوا كل طرف كيف كاتب

## Listening for UDP Packets (Cont.)



like dataInputStream, BufferedInputStream...



أقرأ الخط بترجيبة لربيب

(reverse operation) كشافة نطلع الـ data صحيحة وواضحة.

يفقد اركب عليه Filter Reader BufferReader زي



Eng. Asma Abdel Karim  
Computer Engineering Department

26

# Sending UDP Packets

← نفس ال Socket استخدمنا

- The same interface (*DatagramSocket*) employed to receive UDP packets is also used to send them.
- When sending a packet, the application must create a *DatagramPacket*, set the address and port information, and write the data intended for transmission to its byte array.
- If replying to a received packet, the address and port information will already be stored, and only the data need be overwritten. → بدوننا ننشئ Packet ثانية
- Once the packet is ready for transmission, the send method of *DatagramSocket* is invoked, and a UDP packet is sent.

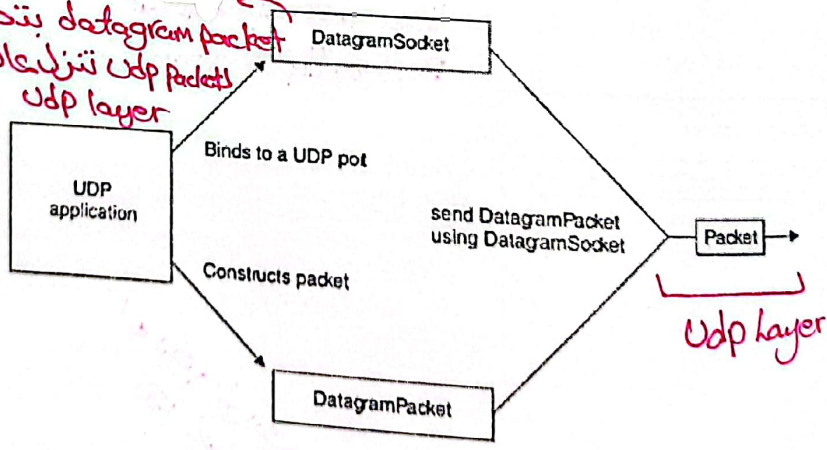
← مسؤولية تعبیه

```
MySocket.send(MyPacket)
```

← Listing ال Socket

# Sending UDP Packets (Cont.)

← بيقولنا Datagram packet بتعملها ال UDP layer  
UDP packets تنزلها ال UDP layer



## Sending UDP Packets (Cont.)

```

DatagramSocket socket = new DatagramSocket(2000);
DatagramPacket packet = new DatagramPacket (new byte[256], 256);
packet.setAddress ( InetAddress.getByAddress ( "somehost" ) );
packet.setPort ( 2000 );
boolean finished = false;
while ( !finished )

```

```

{
    // Write data to packet buffer
    .....
    socket.send ( packet );
    // Do something else, like read other packets, or check to
    // see if no more packets to send
    .....
}

```

عشان يكون  
عملية الكتابة  
أفضل بركب  
Filter streams.....

```

socket.close();

```

ليه يدخل يبعث بـ data مختلفة  
لنفس الـ receiver

هدول الـ 3 أسطر  
بختصوهم بغير بسطر واحد .  
هذا مفيد لما يكون كلامك بيك  
تبعث لعدة مختلف  
فبفصلوهم عنا  
بعض .



Eng. Asma Abdel Karim  
Computer Engineering Department

29

## Additional Information on UDP

- While the UDP is sometimes the best alternative for certain classes of applications, because of its unique properties, it does present some challenges to developers.

- ① – Lack of guaranteed delivery.
- ② – Lack of guaranteed packet sequencing.
- ③ – Lack of flow control.

لو بولك بكمية كبيرة بيضل  
يتراف يتصرف .

الـ data ممكن توصل  
out of ordering



Eng. Asma Abdel Karim  
Computer Engineering Department

30

## References

Chapter 5 of *Java™ Network Programming and Distributed Computing*, David Reilly and Michael Reilly.



Eng. Asma Abdel Karim  
Computer Engineering Department

31



# Networks and Internet Programming

## Transmission Control Protocol



Eng. Asma Abdel Karim  
Computer Engineering Department

1

## Outline

- Overview. ✓
- Advantages of TCP Over UDP. ✓
- Communication between Applications Using Ports. ✓
- Socket Operations. ✓
- TCP and the Client/Server Paradigm. ✓
- TCP Sockets and Java. ✓
- Socket Class. ✓
- Creating a TCP Client. ✓
- ServerSocket Class. ✓
- Creating a TCP Server. ✓
- Exception Handling: Socket-Specific Exceptions. ✓



Eng. Asma Abdel Karim  
Computer Engineering Department

1

## Overview

بكون أفضل وأخف من ال UDP ←

- The properties of TCP make it highly attractive to network programmers.
  - As it simplifies network communication by removing many of the obstacles of UDP, such as ordering of packets and packet loss.
- UDP is concerned with the transmission of packets of data.
  - TCP focuses instead on establishing a network connection, through which a stream of bytes may be sent and received.

العائق الذي يتسبب مشاكل زي ال UDP ال الذي يساهم في حلها.

\* بال TCP إنشاء ال Socket يتضمن عملية ال handshaking مع الطرف المقابل



Eng. Asma Abdel Karim  
Computer Engineering Department

to create virtual connection

و بيحتاجه input stream و output stream

## Overview (Cont.)

- Packets may be sent through a network using various paths and may arrive at different times. → out of order ممكن يوصلوا
- This benefits performance and robustness, as the loss of a single packet doesn't necessarily disrupt the transmission of other packets.
- Nonetheless, such a system creates extra work for programmers who need to guarantee delivery of data.
- TCP eliminates this extra work by guaranteeing delivery and order, providing for a reliable byte communication stream between client and server that supports two-way communication. → 2 streams (input / output)
- TCP establishes a "virtual connection" between two machines, through which streams of data may be sent.

اللي ما يوصلوا مش خراب الكل وبتأخر

هو اللي بيحتاج الضمانات لكل الأمور اللي عليك (extra work)

و الطرف المقابل يكون عكسه



Eng. Asma Abdel Karim  
Computer Engineering Department



\* ال Socket UDP يفتح بال local على الجهاز ما يتضمن أي عملية Communication مع أي طرف ثاني. لكن ال TCP ال Socket فيو بجوي Communication

لازم نفتح Socket بال TCP

## Overview (Cont.)

- \* ① The virtual connection between two machines is represented by a socket.
- There are substantial differences between a UDP socket and a TCP socket.
  - ② - First, TCP sockets are connected to a single machine, whereas UDP sockets may transmit or receive data from multiple machines.
  - ③ - Second, UDP sockets only send and receive packets of data, whereas TCP allows transmission of data through byte streams (represented as an InputStream and OutputStream). They are converted into datagram packets for transmission over the network, without requiring the programmer to intervene.

Convert (بواسطة API) lower layer بتعمل هذا بال

Eng. Asma Abdel Karim  
Computer Engineering Department

ال Socket ال TCP من ال UDP ال ① ② ③  
 Connect او استخدمت وهو دايم باستخدامه  
 بدون ما انت تدخل بعملية  
 تحيد ال Streams ال Packets

## Advantages of TCP over UDP

- Automatic Error Control.
- Reliability.
- Ease of Use.

Eng. Asma Abdel Karim  
Computer Engineering Department

8

## Communication between Applications Using Ports

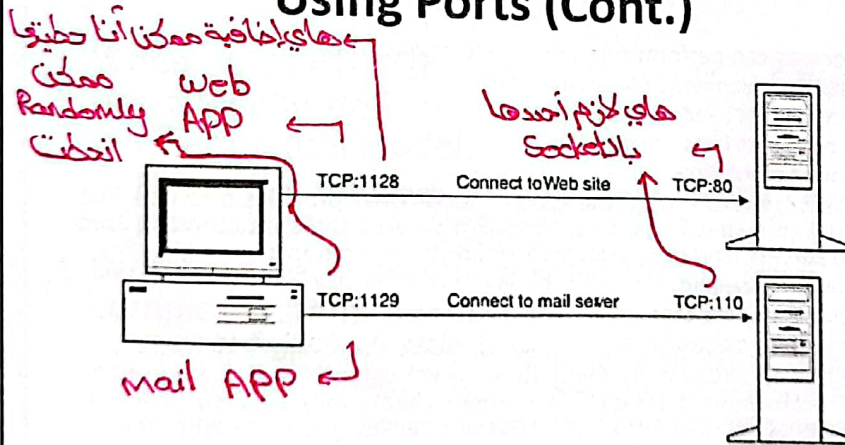
- It is clear that there are significant differences between TCP and UDP, but there is also an important similarity between these two protocols. Both share the concept of a communications port, which distinguishes one application from another.
- When a TCP socket establishes a connection to another machine, it requires two very important pieces of information to connect to the remote end—the IP address of the machine and the port number.
- In addition, a local IP address and port number will be bound to it, so that the remote machine can identify which application established the connection.

ال Socket ل  
 ال TCP و ال UDP  
 اللذين يشتركا  
 باستخدام  
 ال Ports للتمييز  
 مع مينا تتواصل  
 دائما هال  
 ال Port لازم  
 نعرفهم  
 (تأكد الطرف  
 الثاني)

لـ بالإضافة ممكننا نعطيني ال ال local لو بدك ،  
 ولو ما انطقوا بنحطوا Randomly عاينى

\* اللى بيخالف بال TCP من ال socket كذا انك لازم تحدد معلومات الطرف الثاني.. \*

## Communication between Applications Using Ports (Cont.)



## Communication between Applications Using Ports (Cont.)

- Ports in TCP are just like ports in UDP—they are represented by a number in the range 1-65535.
- Ports below 1024 are restricted to use by well-known services such as HTTP, FTP, SMTP, POP3, and telnet.

Port Num  
→ 16 bit

Well-Known Services	Service Port
Telnet	23
Simple Mail Transfer Protocol	25
HyperText Transfer Protocol	80
Post Office Protocol 3	110

Eng. Asma Abdel Karim  
Computer Engineering Department

كل اللي تحت 1024 محجوزين

APP معروفه ، لو انا جمل ال APP تايه بنقيا  
Port Number فوق 1024

Class ال  
اللي بعد ال  
Virtual connection  
اللي  
Socket

## Socket Operations

- TCP sockets can perform a variety of operations. They can:
  - Establish a connection to a remote host.
  - Send data to a remote host.
  - Receive data from a remote host.
  - Close a connection.
- In addition, there is a special type of socket that provides a service that will bind to a specific port number. This type of socket is normally used only in servers, and can perform the following operations:
  - Bind to a local port.
  - Accept incoming connections from remote hosts.
  - Unbind from a local port.
- These two sockets are grouped into different categories, and are used by either a client or a server (since some clients may also be acting as servers, and some servers as clients). However, it is normal practice for the role of client and server to be separate.

Server Socket  
Does not represent connection  
local بعد ال  
Server socket  
ال Server ال  
Binding بعد ال  
local Port S  
ويبلش يسع ال  
incoming connections  
requests from client  
and accept them

عده بنستخدمه الثاني  
بالعادة بكونا معروف وواضح  
Client و Server مين ال

Eng. Asma Abdel Karim  
Computer Engineering Department

لكل accept بفتح Beck connection مع ال client ولما ينقيا  
يعمل unbind ال local ports

Eng. Asma Abdelkarim

# TCP and the Client/Server Paradigm

- The client/server paradigm divides software into two categories, clients and servers.
- \* - A client is software that initiates a connection and sends requests, whereas →
- \* - A server is software that listens for connections and processes requests. →
- In the context of UDP programming, no actual connection is established, and UDP applications may both initiate and receive requests on the same socket.
- In the context of TCP, where connections are established between machines, the client/server paradigm is much more relevant.

Eng. Asma Abdel Karim  
Computer Engineering Department

Peer-to-Peer Client-Server  
كل بيختم كالم (Server) كل بيختم كالم (Client)

لا يوجد اتصال Socket  
من نوع واحد

+ كل عينة  
Connection  
Socket  
كل Socket  
Connection  
Request  
Server - Client  
أكثر

# TCP and the Client/Server Paradigm (Cont.)

- When software acts as a client, or as a server, it has a rigidly defined role that fits easily into a familiar mental model.
- \* - Either the software is initiating requests, or it is processing them. →
- Switching between these roles makes for a more complex system.
- \* - Even if switching is permitted, at any given time one software program must be the client and one software program must be the server. If they both try to be clients at the same time, no server exists to process the requests!

Eng. Asma Abdel Karim  
Computer Engineering Department

when switching, Client  
at any time → Server  
منه التين نفس الاتي

14

ال Server مبدئي

حسب سؤال Client يطلب ال Server العمل

ال Client الذي يقدر يتحكم فيها

# Network Clients

- \* Network clients initiate connections and usually take charge of network transactions.
- \* The server is there to fulfill the requests of the client—a client does not fulfill the requests of a server.
- \* Although the client is in control, some power still resides in the server, of course. A client can tell a server to delete all files on the local file system, but the server isn't necessarily compelled to carry out that action.
- \* The network client speaks to the server using an agreed-upon standard for communication, the network protocol.
  - For example, an HTTP client uses a set of commands different from a mail client, and has a completely different purpose.
  - Connecting an HTTP client to a mail server, or a mail client to an HTTP server, will result not only in an error message but in an error message that the client will not understand.
  - For this reason, as part of the protocol specification, a port number is used so that the client can locate the server.

نحن فاهنا ال ch ما نعمل هيك متيزل قوي  
ع ال Top فوظيفة ا خطي

التفقا بين ال Server وال Client

# Network Servers

لازم يكون Infinite loop دائما.  
ال Server  
لانها لازم يكون  
تغال لاي وقت  
ممكنا يجي فيه  
Client

- = The role of the network server is to bind to a specific port (which is used by the client to locate the server), and to listen for new connections. *مؤقت*
- = While the client is temporary, and runs only when the user chooses, the server must run continually (even if no clients are actually connected) in the hope that someone, at some time, will want its services. *Single threaded*
- = Some servers can handle only a single connection at a time, while others can handle many connections concurrently, through the use of threads.

لو ابا Client طلبنا ال Server وهو كان لسا بخدم Client ثاني،  
رح يقعد بخدمه بيشي ال Client الاول ليخلص.

multithreaded بغير كانه عندي اكثر من Servers تحت

ال Server الاولية يتبعها اكثر من

Client



## TCP Sockets and Java

- Java offers good support for TCP sockets, in the form of two socket classes, java.net.Socket and java.net.ServerSocket.
- When writing client software that connects to an existing service, the Socket class should be used.
- When writing server software that binds to a local port in order to provide a service, the ServerSocket class should be employed.
- This is different from the way a DatagramSocket works with UDP.
- \* - The function of connecting to servers, and the function of accepting data from clients, is split into a separate class under TCP.

جهة ال Server  
هو اللي بيلش فتحة ال Connection وفتحها هي انشاء ال Socket

Eng. Asma Abdel Karim  
Computer Engineering Department

17

Socket back مع الطرف الثاني

Communication Virtual channel

## Socket Class

- The Socket class represents client sockets, and is a communication channel between two TCP communications ports belonging to one or two machines.
- A socket may connect to a port on the local system, avoiding the need for a second machine, but most network software will usually involve two machines.
- TCP sockets can't communicate with more than two machines, however.
- \* - If this functionality is required, a client application should establish multiple socket connections, one for each machine.

غالباً نفس الجهاز يكون مشاننا  
اعمل testing لـ APP

ال فرق عن ال UDP

\* بال TCP كل ال devices اللي عنك بتحتاج تفتح Socket ال

Eng. Asma Abdel Karim  
Computer Engineering Department

18

Socket الومية بقدر ادهي عن طريقها مع device ثاني فقط لا غير.

أنا و Socket  
 انهم موافق هذا ال  
 Connection  
 ال object يعمل  
 والأمر ينتمي أنا  
 ما و كل رد إنه موافق  
 Socket  
 Exception

## Socket Class - Constructors

أسهل وحدة التي يتخط فيها معلومات الطرف الثاني.

- The easiest way to create a socket is to specify the hostname of the machine and the port of the service. For example, to connect to a Web server on port 80, the following code might be used:

```
try{
    // Connect to the specified host and port
    Socket mySocket = new Socket ("www.awl.com", 80);
    // .....
}
catch (Exception e){
    System.err.println ("Err - " + e);
}
```

يعمل DNS كثنان  
 يقطي ال IP Address

لازم أعطي  
 معلومات

ال destination مش  
 الطرف الثاني  
 زي ال domain ما ينحط معلومات

Eng. Asma Abdel Karim  
 Computer Engineering Department

يا يعمل  
 rejection  
 ال Connection  
 ال Connection  
 ما يمكن وحدة  
 خلال ال timeout

الطرف الثاني

## Socket Class – Constructors (Cont.)

- \* **protected Socket ()**
  - Creates an unconnected socket using the default implementation provided by the current socket factory. Developers should not normally use this constructor, as it does not allow a hostname or port to be specified.
- \* **Socket (InetAddress address, int port)** throws **java.io.IOException, java.lang.SecurityException**
  - Creates a socket connected to the specified IP address and port. If a connection cannot be established, or if connecting to that host violates a security restriction (such as when an applet tries to connect to a machine other than the machine from which it was loaded), an exception is thrown.
- \* **Socket (InetAddress address, int port, InetAddress localAddress, int localPort)** throws **java.io.IOException, java.lang.SecurityException**
  - Creates a socket connected to the specified address and port, and is bound to the specified local address and local port. By default, a free port is used, but this method allows you to specify a specific port number, as well as a specific address, in the case of multihomed hosts (i.e., a machine where the localhost is known by two or more IP addresses).

الهدف منه إنه ال Subclasses يستفيدوا منه  
 مش لإنشاء ال object

ما حستجوه  
 معلومات  
 الطرف الثاني فقط  
 ال ال Codes  
 الموافقين  
 لد طعم  
 الأيكر  
 ال ال App  
 مو مسؤولة  
 ينشئ Socket  
 مثل  
 ال applet

التعدي  
 Socket  
 Exception

حدثت معلوماتي ومعلومات الطرف الثاني

Eng. Asma Abdel Karim  
 Computer Engineering Department

Eng. Asma Abdelkarim

## Socket Class -- Constructors (Cont.)

- ما حستخدام
- \* protected Socket (SocketImpl implementation)
    - Creates an unconnected socket using the specified socket implementation. Developers should not normally use this constructor, as it does not allow a hostname or port to be specified.
  - \* Socket (String host, int port) throws java.io.IOException, java.lang.SecurityException, java.net.UnknownHostException
    - Creates a socket connected to the specified host and port. This method allows a string to be specified, rather than an InetAddress. If the hostname could not be resolved, a connection could not be established, or a security restriction is violated, an exception is thrown.
  - \* Socket (String host, int port, InetAddress localAddress, int localPort) throws java.net.UnknownHostException, java.io.IOException, java.lang.SecurityException
    - Creates a socket connected to the specified host and port, and bound to the specified InetAddress instance, as well as allowing a specific local address and port to be bound to. These local parameters are useful for multihomed hosts (i.e., a machine where the localhost is known by two or more IP addresses). If the hostname can't be resolved, a connection cannot be established, or a security restriction is violated, an exception is thrown.

الفرق بينهم وبين  
الاثني الذي قبل  
انه بتطبيقه اد  
Name & Address  
وبعد  
DNS lookup  
فان  
UnknownHostException  
لو فشلت  
على ال DNS

## Creating a Socket

- = Under normal circumstances, a socket is connected to a machine and port when it is created.
- \* Although there is a blank constructor that does not require a hostname or port, it is protected and can't be called from normal applications.
  - Furthermore, there isn't a connect() method that allows you to specify these details at a later point in time, so under normal circumstances the socket will be connected when created.
- = If the network is fine, the call to a socket constructor will return as soon as a connection is established, but if the remote machine is not responding, the constructor method may block for an indefinite amount of time.
- \* This varies from system to system, depending on a variety of factors such as the operating system being used and the default network timeout. → يحدد بال TCP
- \* In mission-critical systems it may be appropriate to place such calls in a second thread, to prevent an application from stalling.

كيفية Socket تشيئة  
جهاز و Port  
معين لما تعمل  
بس يكون  
بال Subclass  
بانشاء

منها ال TCP يتشبه  
وتنقل الزمان  
قادرة تشيئة ال  
Socket لانه من  
قادرة تعمل ال  
Virtual connection  
بشكل من System  
System

عشان لو عرفت على ان ثانية ما بتعتمد على هاي  
العملية الأفضل نعملهم بـ thread ثانية  
عشان ما تقفل علوم

في Methods بتغير بال option وفي Methods بتعمل Setter و Getter.  
 \* ما تطلب بال options ناسك ال Socket ال لو كنت ظلم شو بتعمل \*

أما لو في data معلقة ما اجاني acknowledge عليها ممكن تسبقت أو لا حسب أنواع ال Stream المستخدمة.

## Using a Socket

لما أسكر ال Socket إذا في data عالقة بال buffer بتسبقت أكيد

- void **close()** throws java.io.IOException  
 - Closes the socket connection. Closing a connect may or may not allow remaining data to be sent, depending on the streams before closing a socket connection.
- InetAddress **getInetAddress()**  
 - Returns the address of the remote machine that is connected to the socket. الطرف الثاني
- InputStream **getInputStream()** throws java.io.IOException  
 - Returns an input stream, which reads from the application this socket is connected to.
- OutputStream **getOutputStream()** throws java.io.IOException  
 - Returns an output stream, which writes to the application that this socket is connected to.

Eng. Asma Abdel Karim  
 Computer Engineering Department

\* الوم كل وحدة شرح بسلايين لقادم \*

23

يشمل إزفاء ال Virtual connection مع الطرف الثاني

Socket options :

## Using a Socket (Cont.)

- boolean **getKeepAlive()** throws java.net.SocketException  
 - Returns the state of the SO\_KEEPALIVE socket option.
- InetAddress **getLocalAddress()**  
 - Returns the local address associated with the socket (useful in the case of multihomed machines).
- int **getLocalPort()**  
 - Returns the port number that the socket is bound to on the local machine.
- int **getPort()**  
 - Returns the port number of the remote service to which the socket is connected.
- int **getReceiveBufferSize()** throws java.net.SocketException  
 - Returns the receive buffer size used by the socket, determined by the value of the SO\_RCVBUF socket option.
- int **getSendBufferSize()** throws java.net.SocketException  
 - Returns the send buffer size used by the socket, determined by the value of the SO\_SNDBUF socket option.

Eng. Asma Abdel Karim  
 Computer Engineering Department

هناك غير ال buffers البيانات بتعمل ال Streams

بتقعد فيه ال data الطالعة .

24

ببجولي تابعي الطرف الثاني  
 ال buffer بتقعد فيه ال data ال الامة هاي ال buffer موجود جوا ال Socket

## Using a Socket (Cont.)

- **int getSoLinger()** throws `java.net.SocketException`
  - Returns the value of the `SO_LINGER` socket option, which controls how long unsent data will be queued when a connection is terminated.
- **int getSoTimeout()** throws `java.net.SocketException`
  - Returns the value of the `SO_TIMEOUT` socket option, which controls how many milliseconds a read operation will block for. If a value of 0 is returned, the timer is disabled and a thread will block indefinitely (until data is available or the stream is terminated).
- **boolean getTcpNoDelay()** throws `java.net.SocketException`
  - Returns "true" if the `TCP_NODELAY` socket option is set, which controls whether Nagle's algorithm is enabled.

بنورجيني  
لو أنا عامل  
limit و blocking  
معيته ، او  
رجولي ه يقيني  
يكون ال default  
(blocking  
forever)



## Using a Socket (Cont.)

- **void setKeepAlive(boolean onFlag)** throws `java.net.SocketException`
  - Enables or disables the `SO_KEEPALIVE` socket option.
- **void setReceiveBufferSize(int size)** throws `java.net.SocketException`
  - Modifies the value of the `SO_RCVBUF` socket option, which recommends a buffer size for the operating system's network code to use for receiving incoming data. Not every system will support this functionality or allows absolute control over this feature. If you want to buffer incoming data, you're advised to instead use a `BufferedInputStream` or a `BufferedReader`.
- **void setSendBufferSize(int size)** throws `java.net.SocketException`
  - Modifies the value of the `SO_SNDBUF` socket option, which recommends a buffer size for the operating system's network code to use for sending incoming data. Not every system will support this functionality or allows absolute control over this feature. If you want to buffer incoming data, you're advised to instead use a `BufferedOutputStream` or a `BufferedWriter`.

اتحكم  
بحجم ال buffer  
تعال receive  
، مو شريك يقبل  
منك هالقرار  
و ممكن يحطلك  
قيود



ال buffering على Socket امن ال buffering على Socket  
ال OS بيحد عليه كثير قيود و هو اوتوب ال TCP.  
\* ال buffering على Streams اوتوب و بيحد عليها بل ال App  
فالك سلطة عليها

## Using a Socket (Cont.)

- `static void setSocketImplFactory (SocketImplFactory _factory)`  
throws `java.net.SocketException`, `java.io.IOException`  
`java.lang.SecurityException`
  - Assigns a socket implementation factory for the JVM, which may already exist, or may violate security restrictions, either of which causes an exception to be thrown. Only one factory can be specified, and this factory will be used whenever a socket is created.
- `void setSoLinger(boolean onFlag, int duration)` throws  
`java.net.SocketException` `java.lang.IllegalArgumentException`
  - Enables or disables the `SO_LINGER` socket option (according to the value of the `onFlag` boolean parameter), and specifies a duration in seconds. If a negative value is specified, an exception is thrown.

← خذني  
كيف بقدين



## Using a Socket (Cont.)

- `void setSoTimeout(int duration)` throws `java.net.SocketException`
  - Modifies the value of the `SO_TIMEOUT` socket option, which controls how long (in milliseconds) a read operation will block. A value of zero disables timeouts, and blocks indefinitely. If a timeout does occur, a `java.io.IOException` is thrown whenever a read operation occurs on the socket's input stream. This is distinct from the internal TCP timer, which triggers a resend of unacknowledged datagram packets.
- `void setTcpNoDelay(boolean onFlag)` throws  
`java.net.SocketException`
  - Enables or disables the `TCP_NODELAY` socket option, which determines whether Nagle's algorithm is used.

← عملية القراءة  
ليتم من الinput stream  
وهي التي بنعملها  
block لمدة زمنية  
معينه



## Using a Socket (Cont.)

- `Socket` ينسجك ال `InputStream` فقط مش كل ال `Socket` ←  
**`void shutdownInput()` throws `java.io.IOException`**
  - Closes the input stream associated with this socket and discards any further information that is sent. Further reads to the input stream will encounter the end of the stream marker.
- **`void shutdownOutput()` throws `java.io.IOException`**
  - Closes the output stream associated with this socket. Any data previously written, but not yet sent, will be flushed, followed by a TCP connection-termination sequence, which notifies the application that no more data will be available (and in the case of a Java application, that the end of the stream has been reached). Further writes to the socket will cause an `IOException` to be thrown.

أي شئ يعنيه الطرف الثاني ما بيوصلني يتم تجلله ولو أنا حاولت اقرأ منه حيطلعلي end of stream لأنني سكرته.

تسكرو output stream فقط مش كل ال Socket

يعني الطرف الثاني بقدر يبعثلي كمان بس نقله ابي. أنا حبطت أقدر أبعت  
 Eng. Asma Abdel Karim  
 Computer Engineering Department

بغير الطرف الثاني أنه وصلت end of stream

29

## Reading from and Writing to TCP Sockets

- Once a socket is created, it is connected and ready to read/write by using the socket's input and output streams.
- These streams don't need to be created; they are provided by the `Socket.getInputStream()` and `Socket.getOutputStream()` methods. *مش صنفه انه ال class يكون static → قصه انه هيلج ال method تبعها ال class*
- A filter can easily be connected to a socket stream, to make for simpler programming. *وبقدر أركب Readers/writers →*

ممكن نقرا منوم نفري بس بتفري low bytes لأنه يكون level

Eng. Asma Abdel Karim  
 Computer Engineering Department

30

## Reading from and Writing to TCP Sockets (Example)

```

try{
    // Connect a socket to some host machine and port
    Socket socket = new Socket ( somehost, someport ); → Socket أنشئ
    // Connect a buffered reader
    BufferedReader reader = new BufferedReader ( new
    InputStreamReader ( socket.getInputStream() ) );
    // Connect a print stream
    PrintStream pstream = new PrintStream(
    socket.getOutputStream() );
}
catch (Exception e){
    System.err.println ("Error - " + e);
}

```

كائن  
 تركيب  
 Reader

هذا بتقري  
 تستخدم  
 Reader.read line

بتقري  
 تستخدم  
 Pstream.Print

or Pstream.println



Eng. Asma Abdel Karim  
Computer Engineering Department

31

## Socket Options

- Socket options are settings that modify how sockets work, and they can affect (both positively and negatively) the performance of applications.
- Generally, socket options should not be changed unless there is a good reason for doing so, as changes may negatively affect application and network performance.
- The one exception to this caveat is the SO\_TIMEOUT option.
  - Virtually every TCP application should handle timeouts gracefully rather than stalling if the application the socket is connected to fails to transmit data when required.



Eng. Asma Abdel Karim  
Computer Engineering Department

32

من القاعدة التي فوق الاستثناء الوحيد هو Timeout هذا الذي  
 لا تغير option تاهة والأفضل  
 انه مانغلقه blocking forever زي ما هو  
 by default



## SO\_KEEPALIVE Socket Option

- By default, no data is sent between two connected sockets unless an application has data to send.
- \* - This means that an idle socket may not have data submitted for minutes, hours, or even days in the case of long-lived processes.
- Suppose, however, that a client crashes, and the end-of-connection sequence is not sent to a TCP server.
  - Valuable resources (CPU time and memory) might be wasted on a client that will never respond.
- When the keepalive socket option is enabled, the other end of the socket is probed to verify it is still active.
  - However, the application doesn't have any control over how often keepalive probes are sent.

Eng. Asma Abdel Karim  
Computer Engineering Department

33

بطل  
موجود بدون  
ما ينوي  
ال Connection  
بيننا فويك  
مزيدوا علينا  
كبوت و overhead  
ويأخذوا من ال Mem  
وال time ، لهيك

SO-KEEPALIVE option كذا لو فعلته بتبصر ال TCP نبعث نوع من ال Messages الهدف منهم التقدر اذا الطرف الثاني كاشيت أو لا. ( Prob / keepAlive Messages ) ولو كان هو

## SO\_KEEPALIVE Socket Option (Cont.)

- To enable *keepalive*, the `Socket.setSoKeepAlive(boolean)` method is called with a value of "true" (a value of "false" will disable it).
  - For example, to enable *keepalive* on a socket, the following code would be used.
- ```
// Enable SO_KEEPALIVE
someSocket.setSoKeepAlive(true);
```
- It should also be kept in mind that *keepalive* doesn't allow you to specify a value for probing socket endpoints.
    - A better solution than *keepalive*, and one that developers are advised to use, is to instead modify the timeout socket option.

Eng. Asma Abdel Karim  
Computer Engineering Department

34

يعني ما بتحكم  
بال Messages Prob  
اللي بيقوم ولا تمتا  
يقدر انه الطرف الثاني  
خسراه ولا كذا متنى  
تبعث ال Prob Msg.

ما بتحل مشكلة الطرف الثاني انه  
مو Alive فالطرف الثاني انه

بس بتسولنا الحل بديل ما بتستخدم ال SO-KEEPALIVE  
أو تحلوا ع مشوي الحل .

2

## SO\_RCVBUF Socket Option

- The receive buffer socket option controls the buffer used for receiving data.
- Changes can be made to the size by calling the `Socket.setReceiveBufferSize(int)` method.
- For example, to increase the receive buffer size to 4,096 bytes, the following code would be used.

```
// Modify receive buffer size  
someSocket.setReceiveBufferSize(4096);
```



## SO\_RCVBUF Socket Option (Cont.)

- Note that a request to modify the size of the receive buffer does not guarantee that it will change. → لأنه التحكم حسب ال OS
- For example, some operating systems may not allow this socket option to be modified, and will ignore any changes to the value.
- The current buffer size can be determined by invoking the `Socket.getReceiveBufferSize()` method.
- A better choice for buffering is to use a `BufferedReader`.

لا ندم بكوننا عسوفى  
App في الي نتحكم فيهم افضل



## 3) SO\_SNDBUF Socket Option

نفس ال RCVBUF - SO بس هاد بقدر فيه ال ماتفه اللي طالعة

The send buffer socket option controls the size of the buffer used for sending data.

By calling the `Socket.setSendBufferSize(int)` method, you can attempt to change the buffer size, but requests to change the size may be rejected by the operating system.

```
//Set the send buffer size to 4096 bytes
someSocket.setSendBufferSize(4096);
```

To determine the size of the current send buffer, you can call the `Socket.getSendBufferSize()` method, which returns an int value.

```
// Get the default size
int size = someSocket.getSendBufferSize();
```

## 4) SO\_LINGER Socket Option

When a TCP socket connection is closed, it is possible that data may be queued for delivery and not yet sent (particularly if an IP datagram becomes lost in transit and must be resent).

The linger socket option controls the amount of time during which unsent data may be sent, after which it is discarded completely.

It is possible to enable/disable the linger option entirely, or to modify the duration of a linger, by using the `Socket.setSoLinger(boolean onFlag, int duration)` method:

```
// Enable linger, for fifty seconds
```

```
someSocket.setSoLinger(true, 50);
```

```
or
```

```
someSocket.setSoLinger(false);
```

لو تسكو ال Connection ولسا في ماتفه مو مبعوثه يا اما عالقة بال buffed او ما وصلوا acknowledgment ففديه انا ك TCP اخذ احوال ابعت هاي ال ماتفه بعد ما ال Connection تيسكر هذا ال SO-LINGER هو اللي بتتكم به الموضع

لو ما بيبي افعولها واخلي الماتفه العالقة تنبنت ل

5

## TCP\_NODELAY Socket Option

Request for comment Document

- This socket option is a flag, the state of which controls whether Nagle's algorithm (RFC 896) is enabled or not.
- Because TCP data is sent over the network using IP datagrams, a fair bit of overhead exists for each packet, such as IP and TCP header information.
- If only a few bytes at a time are sent in each packet, the size of the header information will far exceed that of the data.
- On a local area network, the extra amount of data sent probably won't amount to much, but on the Internet, where hundreds, thousands, or even millions of clients may be sending such packets through individual routers, this adds up to a significant amount of bandwidth consumption.

لما نبيع ال header  
تاعت ال TCP وال IP  
بتتبع كبرق كبير  
لو كانت هيك  
طريق الإيجل بتتبع  
نسبة ال Metadata  
أقل من نسبة ال  
data الفعلية اللي  
بتتبع.  
data إضافية  
مش فعلية.



Eng. Asma Abdel Karim  
Computer Engineering Department

29

ال NODELAY ما بتخليك تفضل تبعت أجزاء صغيرة من ال data  
وكل جزئ تركله Metadata لا ، بتخليك تستنى لنطول ال data حجم  
معين بعدين بتقل ال data .

## TCP\_NODELAY Socket Option (Cont.)

- The solution is Nagle's algorithm, which states that TCP may send only one datagram at a time.
- When an acknowledgment comes back for each IP datagram, a new packet is sent containing any data that has been queued up.
- This limits the amount of bandwidth being consumed by packet header information, but at a not insignificant cost—network latency.
- Since data is being queued, it isn't dispatched immediately, so systems that require quick response times such as X-Windows or telnet are slowed.
- Disabling Nagle's algorithm may improve performance, but if used by too many clients, network performance is reduced.

بخليه مرة وحدة  
يكون كصية  
مخزونة ال data  
تعمل  
ال TCP\_NODELAY  
بحسب حيز ال  
ال Packet ال  
يحل هو ال  
لي قبلها ، انه  
تمام ووصل .

لا م تفضل  
ال TCP  
لتقل  
ال data



Eng. Asma Abdel Karim  
Computer Engineering Department

40

هاي الامثلة بتجيب بتقل ال packets المنارة فتقل ال bandwidth  
لكن بتزيد التكلفة ووقت الانتظار (latency)

(response time) فورا ممكن يضربنا  
خاصة ال App يحتاج quick response  
مثل اللي بتكتب فيها commands .

Eng. Asma Abdelkarim

# TCP\_NODELAY Socket Option (Cont.)

- Nagle's algorithm is enabled or disabled by invoking the `Socket.setTcpNoDelay(boolean state)` method.
- For example, to deactivate the algorithm, the following code would be used:

```
// Disable Nagle's algorithm for faster response times
someSocket.setTcpNoDelay(false); → by default
```

- To determine the state of Nagle's algorithm and the TCP\_NODELAY flag, the `Socket.getTcpNoDelay()` method is used:

```
// Get the state of the TCP_NODELAY flag
boolean state = someSocket.getTcpNoDelay();
```

← لا ترفاه هي enable  
• disable أو



أنا ما يوجد حجم ال datagram يتحدد مستوى ال TCP ، أنا بس يتحدد  
أصله enable أو disable .

## 6 SO\_TIMEOUT Socket Option → أكثر واحد useful

- This timeout option is the most useful socket option.
- By default, I/O operations (be the file- or network-based) are blocking.
- An attempt to read data from an InputStream will wait indefinitely until input arrives.
- If the input never arrives, the application stalls and in most cases becomes unusable (unless multithreading is used).
- A more robust application will anticipate such problems and take corrective action.

ال TCP ممكن بتعطيل كمانه انه ال data توصل بس ممكن  
تحتاج وقت طويل جدًا لو يك الأخطل نفس timeout لوصل ال data



لو كنت عالمة  
 catch  
 IOException  
 على بي بي بي  
 error  
 Super class  
 class  
 لكن الأخطاء  
 catch  
 أرنب وأفضل.

## SO\_TIMEOUT Socket Option (Cont.)

- When the SO\_TIMEOUT option is enabled, any read request to the InputStream of a socket starts a timer.
- When no data arrives in time and the timer expires, a *java.io.InterruptedIOException* is thrown, which can be caught to check for a timeout.
- What happens then is up to the application developer— a retry attempt might be made, the user might be notified, or the connection aborted. → طريقة التعامل في حالة عدم الوصول يعتمد على المبرمج
- The duration of the timer is controlled by calling the *Socket.setSoTimeout(int)* method, which accepts as a parameter the number of milliseconds to wait for data.

## SO\_TIMEOUT Socket Option (Cont.)

- For example, to set a five-second timeout, the following code would be used:
 

```
// Set a five second timeout
someSocket.setSoTimeout ( 5 * 1000 );
```
- Once enabled, any attempt to read could potentially throw an *InterruptedIOException*, which is extended from the *java.io.IOException* class.
- Since read attempts can already throw an *IOException*, no further code is required to handle the exception
  - However, some applications may want to specifically trap timeout-related exceptions, in which case an additional exception handler may be added.

## SO\_TIMEOUT Socket Option (Cont.)

```

try{
    Socket s = new Socket (...);
    s.setSoTimeout ( 2000 );
    // do some read operation ....
}
catch (InterruptedException iioe){
    timeoutFlag = true; // do something special like set a flag
}
catch (IOException ioe){
    System.err.println ("IO error " + ioe);
    System.exit(0);
}

```

هون بحسب ال  
blocking  
واحتماية  
Exception ال



## SO\_TIMEOUT Socket Option (Cont.)

- To determine the length of the TCP timer, the `Socket.getSoTimeout()` method, which returns an int, can be used.   
↳ milliseconds بنوعه عدد ال
- A value of zero indicates that timeouts are disabled, and read operations will block indefinitely.

```

// Check to see if timeout is not zero
if ( someSocket.getSoTimeout() == 0)
    someSocket.setSoTimeout (500);

```



## Creating a TCP Client

```

import java.net.*;
import java.io.*;
public class DaytimeClient{
    public static final int SERVICE_PORT = 13;
    public static void main(String args[]){
        // Check for hostname parameter
        if (args.length != 1){
            System.out.println ("Syntax - DaytimeClient host");
            return;
        }
        // Get the hostname of server
        String hostname = args[0];
        try{
            // Get a socket to the daytime service
            Socket daytime = new Socket (hostname, SERVICE_PORT);

```

يرجع التاريخ والوقت

تاع ال server  
مش تاعوي

لايني  
هو عكس  
الجوان

بخط ال loopback  
لو عكسه

هاي ال  
Statement  
اللي بتفتح



Eng. Asma Abdel Karim  
Computer Engineering Department

تكون ال Server (remote machine)  
معلوماتي حتمناخذ Randomly لايني هو محددة أنا.  
وتنتظر القبول

Server Socket  
لو اشتغلت  
يعني انقبلت  
هاي ال Socket  
وانفتح ال  
Connection.

## Creating a TCP Client

```

System.out.println ("Connection established");
// Set the socket option just in case server stalls
daytime.setSoTimeout ( 2000 );
// Read from the server
BufferedReader reader = new BufferedReader (
    new InputStreamReader(daytime.getInputStream()));
System.out.println ("Results : " +reader.readLine());
// Close the connection
daytime.close();
catch (IOException ioe){
    System.err.println ("Error " + ioe);
}
}
}

```

2 seconds  
لعملية القراءة

لو خلال  
ثانيتين  
ما واطني  
ال  
date  
وال time

لازم اقبل  
text  
لايني كحتت  
text  
بال server

InterruptedIOException  
بس هون شاملينه بال IOException



Eng. Asma Abdel Karim  
Computer Engineering Department



## ServerSocket Class

- A special type of socket, the *server socket*, is used to provide TCP services.
- Client sockets bind to any free port on the local machine, and connect to a specific server port and host.
- The difference with server sockets is that they bind to a specific port on the local machine, so that remote clients may locate a service.
- Client socket connections will connect to only one machine, whereas server sockets are capable of fulfilling the requests of multiple clients.

↓  
 هي لا تتقبل Connection مع Client مع connection with request  
 ServerSocket مع Client مع Socket back  
 Client (كلها لو قبلت ال connection)  
 ServerSocket زي Socket



Eng. Asma Abdel Karim  
 Computer Engineering Department

49

## ServerSocket Class (Cont.)

- Clients are aware of a service running on a particular port.
- Clients establish a connection, and within the server, the connection is accepted.
- \* - Multiple connections can be accepted at the same time, or a server may choose to accept only one connection at any given moment. → Multithreading
- Once accepted, the connection is represented as a normal socket, in the form of a *Socket* object. → ServerSocket ال بيتسوقا
- This *ServerSocket* object acts as a factory for client connections, you don't need to create instances of the *Socket* class yourself.
- \* - These connections are modeled as a normal socket, so you can connect input and output filter streams (or even a reader and writer) to the connection.



Eng. Asma Abdel Karim  
 Computer Engineering Department

50

# Creating a ServerSocket

- Once a server socket is created, it will be bound to a local port and ready to accept incoming connections.
- When clients attempt to connect, they are placed into a queue. Once all free space in the queue is exhausted, further clients will be refused.
- The simplest way to create a server socket is to bind to a local address, which is specified as the only parameter, using a constructor.

```
try{
    // Bind to port 80, to provide a TCP service (like HTTP)
    ServerSocket myServer = new ServerSocket ( 80 );
    // .....
}
catch (IOException ioe){
    System.err.println ("I/O error - " + ioe);
}
```

"local port"

رج يوقفوا  
 Clients ال  
 بال queue  
 يمشو ال Server  
 يخلص شغله  
 مع ال Client  
 اللي يمشو له  
 لو تقبل ال Server  
 للفد خلص بطل  
 يمشو لكان  
 Connection  
 لعمام rejection  
 ويقلم انما انصحت  
 ال Socket  
 تاخوم .

# Creating a ServerSocket - Constructors

- ServerSocket(int port)** throws java.io.IOException, java.lang.SecurityException
  - \* - Binds the server socket to the specified port number, so that remote clients may locate the TCP service.
  - \* - If a value of zero is passed, any free port will be used. However, clients will be unable to access the service unless notified somehow of the port number. *recommended*
  - \* - By default, the queue size is set to 50 but an alternate constructor is provided that allows modification of this setting. *like applets*
  - \* - If the port is already bound, or security restrictions (such as security polices or operating system restrictions on well-known ports) prevent access, an exception is thrown. *queue size 50*
- ServerSocket(int port, int numberOfClients)** throws java.io.IOException, java.lang.SecurityException
  - \* - Binds the server socket to the specified port number and allocates sufficient space to the queue to support the specified number of client sockets.
  - \* - If the port is already bound or security restrictions prevent access, an exception is thrown.

لو طيبه  
بغناجه  
randomly

## Creating a ServerSocket – Constructors (Cont.)

- `ServerSocket(int port, int numberOfClients, InetAddress address)` throws `java.io.IOException`, `java.lang.SecurityException`
- \* – Binds the server socket to the specified port number, and allocates sufficient space to the queue to support the specified number of client sockets.
- \* – This is an overloaded version of the `ServerSocket(int port, int numberOfClients)` constructor that allows a server socket to bind to a specific IP address, in the case of a multihomed machine.
- \* – For example, a machine may have two network cards, or may be configured to represent itself as several machines by using virtual IP addresses.
- \* – Specifying a null value for the address will cause the server socket to accept requests on all local addresses.
- \* – If the port is already bound or security restrictions prevent access, an exception is thrown.

→ local  
لو كان عني  
اكثر من IP  
على Machine  
فيجد بين  
استخدم بيل  
ما ينجوها  
random



Eng. Asma Abdel Karim  
Computer Engineering Department

53

## Using a ServerSocket

- While the `Socket` class is fairly versatile, and has many methods, the Server Socket class doesn't really do that much, other than accept connections and act as a factory for Socket objects that model the connection between client and server.
- The most important method is the accept() method, which accepts client connection requests, but there are several others that developers may find useful.

ما فيها  
كثير تفصيل  
لا تواس  
لتستقبل  
requests



Eng. Asma Abdel Karim  
Computer Engineering Department

54

## Using a ServerSocket – Methods

• **Socket** `accept()` throws `java.io.IOException`, `java.lang.SecurityException`

- \* - Waits for a client to request a connection to the server socket, and accepts it.
- \* - This is a blocking I/O operation, and will not return until a connection is made (unless the timeout socket option is set).
- \* - When a connection is established, it will be returned as a `Socket` object. When accepting connections, each client request will be verified by the default security manager, which makes it possible to accept certain IP addresses and block others, causing an exception to be thrown.

\* - However, servers do not need to rely on the security manager to block or terminate connections—the identity of a client can be determined by calling the `getInetAddress()` method of the client socket.

• **void** `close()` throws `java.io.IOException`

- \* - Closes the server socket, which unbinds the TCP port and allows other services to use it.

Method (هم)  
 ما نستعملها ، بتحدد IP's  
 معينة لقبولها واخرين لا.

من جهة ال Server  
 لو ما عيسى ال IP  
 Connection Close  
 وخلص



## Using a ServerSocket – Methods (Cont.)

• **InetAddress** `getInetAddress()`

- \* - Returns the address of the server socket, which may be different from the local address in the case of a multihomed machine (i.e., a machine whose localhost is known by two or more IP addresses).

• **int** `getLocalPort()`

- \* - Returns the port number to which the server socket is bound.

• **int** `getSoTimeout()` throws `java.io.IOException`

- \* - Returns the value of the timeout socket option, which determines how many milliseconds an `accept()` operation can block for. If a value of zero is returned, the `accept` operation blocks indefinitely.

local → ServerSocket  
 local  
 by default = 0  
 Blocking forever



## Using a ServerSocket – Methods (Cont.)

- ***void implAccept(Socket socket) throws java.io.IOException***
  - This method allows ServerSocket subclasses to pass an unconnected socket subclass, and to have that socket object accept an incoming request.
  - Using the implAccept method to accept the connection, an overridden ServerSocket.accept() method can return a connected socket. Few developers will want to subclass the ServerSocket, and using this should be avoided unless required.
- ***static void setSocketFactory ( SocketImplFactory factory ) throws java.io.IOException, java.net.SocketException, java.lang.SecurityException***
  - Assigns a server socket factory for the JVM. This is a static method, and should be called only once during the lifetime of a JVM. If assigning a new socket factory is prohibited, or one has already been assigned, an exception is thrown.



## Using a ServerSocket – Methods (Cont.)

- ***void setSoTimeout(int timeout) throws java.net.SocketException***
  - \* – Assigns a timeout value (specified in milliseconds) for the blocking accept() operation.
  - \* – If a value of zero is specified, timeouts are disabled and the operation will block indefinitely.
  - \* – Providing timeouts are enabled, however, whenever the accept() method is called a timer starts. When the timer expires, a ***java.io.InterruptedIOException*** is thrown, which allows a server to then take further actions.



\* ال Server لازم يكون دائما listening لل Client requests فعادة يكون با infinite loop .

## Accepting and Processing Requests from TCP Clients

- The most important function of a server socket is to accept client sockets.
- Once a client socket is obtained, the server can perform all the "real work" of server programming, which involves reading from and writing to the socket to implement a network protocol.

// Perform a blocking read operation, to read the next socket connection

```
Socket nextSocket = someServerSocket.accept();
```

// Connect a filter reader and writer to the stream

```
BufferedReader reader = new BufferedReader (new  
InputStreamReader (nextSocket.getInputStream() ) );
```

```
PrintWriter writer = new PrintWriter( new OutputStreamWriter  
(nextSocket.getOutputStream() ) );
```



Eng. Asma Abdel Karim  
Computer Engineering Department

قراءة  
وكتابة  
بين ال  
Server وال Client  
لحالي الى تشغيل

ولو في Clients اخرى انعمم accept بيستوا بال service  
ليطلب هذا ال Client .

## Creating a TCP Server

```
import java.net.*;  
import java.io.*;  
public class DaytimeServer{  
    public static final int SERVICE_PORT = 13;  
    public static void main(String args[]){  
        try{  
            // Bind to the service port, to grant clients access to the TCP daytime  
            //service  
            ServerSocket server = new ServerSocket (SERVICE_PORT);  
            System.out.println ("Daytime service started");  
            // Loop indefinitely, accepting clients  
            for (;;) {  
                // Get the next TCP client  
                Socket nextClient = server.accept();
```



Eng. Asma Abdel Karim  
Computer Engineering Department

60

تتكون  
connected مع ال Client  
لو انقبلت ال connection

بتبليش  
تسمع وتستنى  
لل Client  
هون

Abdelkarim

\* لازم لو واحد حيقر الثاني حيكاتب مش  
التنين يعملوا نفس الاشئ .

## Creating a TCP Server

```
// Display connection details
System.out.println ("Received request from " +
    nextClient.getInetAddress() + ":" + nextClient.getPort() );
// Don't read, just write the message
OutputStream out = nextClient.getOutputStream();
PrintStream pout = new PrintStream (out);
// Write the current date out to the user
pout.print( new java.util.Date() );
// Flush unsend bytes
out.flush();
// Close stream
out.close();
// Close the connection
nextClient.close();
```

للرسالة  
أرسلها  
الخروجية من client  
تقبل  
reader.readLine

تلقى  
ال client

بديا بعد لا client  
date  
وال time

Date object  
with date and  
time of the  
system.

لا buffer  
نقل  
flush

## Creating a TCP Server

```
catch (BindException be){
    System.err.println ("Service already running on port " + SERVICE_PORT);
}
catch (IOException ioe){
    System.err.println ("I/O error - " + ioe);
}
}
```

subclass from IOException

يبرج تحت ال  
Socket  
exception

ينتهي لما تحاول  
تعمل bind على  
Port وال Port  
already taken

لو ما حظيت به انتهى  
بال IOException  
، وايضا حظ الخاص  
بعدين العام عشان  
لو ما انتهى بالخاص  
ينتهي بالعام.

لو علت catch لو ايسبب ذلك الأخطاء التي لها علاقة بال Socket بتتروى هون

### Exception Handling: Socket-Specific Exceptions

- All socket-specific exceptions extend from **SocketException**, so by simply catching that exception, you catch all of the socket-specific ones and write a single generic handler. *عام جدًا بكذا*
- In addition, **SocketException** extends from **java.io.IOException** if you want to provide a catchall for any I/O exception. *لـ بـجـمـير أعم بـزـيـادـة*
- SocketException**
  - The **java.net.SocketException** represents a generic socket error, which can represent a range of specific error conditions. For finer-grained control, applications should catch its subclasses. *لو بدنا نتحكم بشكل أدق نستخدم subclasses*

### Exception Handling: Socket-Specific Exceptions (Cont.)

- BindException** → *بـتـروى لـو الـ Port كان already bound*
  - The **java.net.BindException** represents an inability to bind a socket to a local port. The most common reason for this will be that the local port is already in use.
- ConnectException** → *بـتـروى لـو مـوقـد رـق اعل Connection*
  - The **java.net.ConnectException** occurs when a socket can't connect to a specific remote host and port. There can be several reasons for this, such as that the remote server does not have a service bound to that port, or that it is so swamped by queued connections, it cannot accept any further ones.

بـتـشـمـل حـالـة  
أنه الـ Connection  
وكل الطرف الثاني  
بـسـ ما لـجـا فـي رـد +  
يا إما الـ Connection  
أو لـيـس مـا سـيـر  
و سـجـ



## Exception Handling: Socket-Specific Exceptions (Cont.)

3

### NoRouteToHostException

- هنا يكوننا أصلاً الـ connection ما وصل الطرف الثاني بسبب الشبكة
- \* - The `java.net.NoRouteToHostException` is thrown when, due to a network error, it is impossible to find a route to the remote host.
  - \* - The cause of this may be local (i.e., the network on which the software application is running), may be a temporary gateway or router problem, or may be the fault of the remote network to which the socket is trying to connect. Another common cause of this is that firewalls and routers are blocking the client software, which is usually a permanent condition.

4

### InterruptedIOException

- بنزول لما يحير في الـ timeout
- \* - The `java.net.InterruptedIOException` occurs when a read operation is blocked for sufficient time to cause a network timeout, as discussed earlier in the chapter. Handling timeouts is a good way to make your code more robust and reliable.



## References

Chapter 6 of *Java™ Network Programming and Distributed Computing*, David Reilly and Michael Reilly.



\* معظم الأحيان لا Requests ما يكونوا يعتمدوا بعض فيكون الـ thread  
 thread handle Request ينفذها  
 at the same time.

أنظي البرنامج تاي شغال فيه أكثر من task  
 in Parallel.

# Networks and Internet Programming

## Multi-threading and Parallel Programming

# Outline

- Overview.
- Multi-threading in Java.
- Controlling Threads.
- Threads Priorities.
- Thread Synchronization.
- Inter-thread Communication.

# Overview

- Multi-threaded programming is an important concept in Java networking, as networking clients and servers must often perform several different tasks at a time.
- \* - For example, listening for incoming requests and responses, processing data, and updating the text or graphical user interface for the user.
- It is important for the developer to understand the differences between single-threaded programming, multi-process programming, and multi-threaded programming.

③ ↓  
الي حنطبقه

## Serial execution / Parallel قبلها يطلق Single-Threaded Programming

- Traditional software written in procedural languages is compiled into a machine-readable format, which is called machine code.
- This code is read by a central processing unit (CPU), which executes programming statements one after another, in a sequential manner.
- The time taken to execute each statement may vary (due to the nature of the operation, such as comparing two bytes for equality or adding two numbers together), but until a statement is completed, no further statements will run. This is single-threaded execution.
- The chief advantage of this type of programming is its simplicity.
  - \* Developers can easily predict the state of a machine at any given moment in time. → values of variables.
  - \* It is guaranteed that a variable being accessed in a single-threaded environment will not be accessed or modified by another copy of the program, as only one copy of the program is running.

time ال  
لفوق حسب  
الinstructions

أفضل اشئ فيه  
السوية.

ما في أكثر من  
حد تبشغل بال Var  
تاعتي فيغير بأي  
لحظة أعراف قيمتها

بسوية بدون وجود face condition

مشكلته انه ما تبشغل كل ال cores  
الموجودة بجوازى لا ياشغل بي  
one task بنفس الوقت . فعشنا  
هيك طلع ال multi process/thread

لا يوجد Process يمكن الا Mem / data Code خاص فيها .  
 ال Process ، اخل ال OS على وقت ال Mager execution unit وهو اشي ثقيل

high performance computer.

← أقدم من ال multi-threaded / موجود أكثر بال  
**Multi-process Programming**

\* لينتقل ال Process task وينفذ أكثر من Process بنفس الوقت

- Each application runs as a process, with memory allocated for program code and data storage.
- Multiple processes would run on the same machine.
  - \* - The operating system would allocate CPU time to each process, suspending a process when its time was up and allowing another to take its place.
  - \* - Sometimes, a process will become blocked (waiting on I/O), or may voluntarily choose to yield its CPU time.
  - \* - The operating system creates the illusion that these processes are running concurrently, by frequently switching from one process to another and sharing time between them (though not always equally).
- This type of multitasking is extremely important, as it means that one machine can share its CPU time across many users.

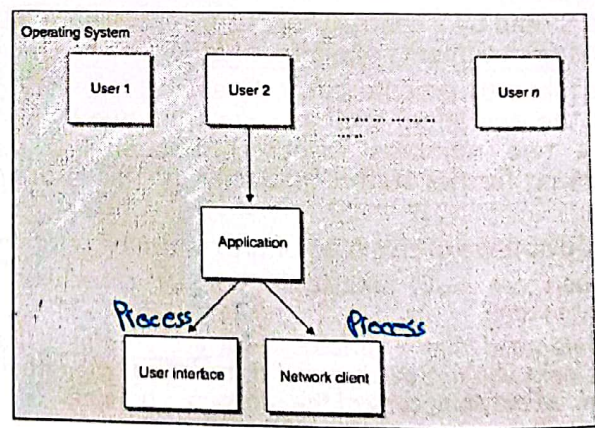
← يعني انجيب ال  
 من سرعة ال task  
 انه يشتغلوا في  
 Parallel  
 بيهم بالأساس  
 ال Process ال  
 يشتغل وينفذ task  
 5 و بعين ال يوجد  
 وهي في قوة ال  
 multi-process  
 (في حال وجود  
 core واحد)

← أقدم من ال Multi-thread Programming

**Multi-process Programming (Cont.)**

لو كنا أكثر من core بقدر يتنفذ أكثر من Process

- Programs themselves could create new processes, having one part of the program performing a task while another part does something else.



## Multi-process Programming (Cont.)

Although multi-process programming works well, there are disadvantages to its use.

First, when a process branches into two, there is overlap between the data storage of one process and another.

- Because two copies of data are being kept, more memory than is needed is consumed.

Second, there isn't an easy way for one process to access and modify the data of another.

\* In Unix, Inter-Process Communication (IPC) is used, creating data pipes that allow a process to communicate with another.

\* Nonetheless, it is not as easy to design software that shares data in a multi-process environment as it is in a multi-threaded one.

لو عملت fork لازم يكون في Mem خاص لكل وحدة حتى لو نفس الdata مشان ما يبيس overlap وهذا استخدام سيبي لـ Mem واذا ختمت على data بتحتاج اشي ثقيل.

التواصل مع كثير  
له اشي ثقيل  
له طلع مشان يحسن  
له لانه لازم توري الثاني انت شو عملت مشان  
يعدل زيكي

## Multi-threaded Programming

Multi-threaded programming requires a different way of looking at software.

Rather than executing a series of steps sequentially, tasks are executed concurrently—that is, many tasks are performed at the same time, rather than one task having to finish before another can start.

Multithreading, also known as multiple threads of execution, allows a program to have multiple instances of itself running, while using the same shared memory space and code.

\* Unlike multi-process programming, which uses separate memory address spaces, making communication between processes difficult.

\* An application can be performing many different tasks concurrently, and threads may access shared data variables to work collaboratively.

اي tasks ما الهم علاقة ببعض  
تتفرق بـ thread  
منطقتين وهيك بتقدر تنفذهم بنفس الوقت.

مشكلة thread  
له اشي خاص لا يكون Sharing نفس ال Mem فويك صار يعب light

واخف وانت بتحدد مين بديك تحط هيك ومين لا

\* لو صنفنا one core بس بتخلى ما تستخدم اشياء لانها مشغل  
one at the same time حتى لو توحيالك من السرعة انهم سوا.

### Multi-threaded Programming (Cont.)

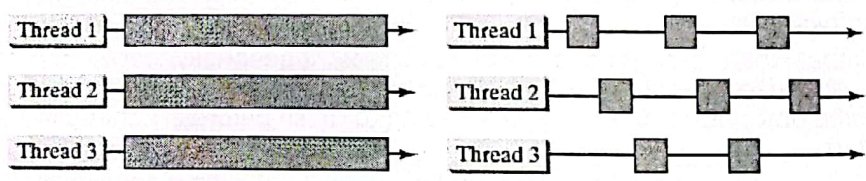
- Unless you have more than one CPU, only a single thread can be running at any given moment in time.
- The operating system maintains a queue of threads and allocates CPU time to them.
- The process of determining which thread to run is called scheduling.
- Not all operating systems allocate thread time fairly, but to give the operating system a guide, threads are allocated a priority level.
- Since the choice of which thread is executed is up to the operating system and not the application, it becomes impossible to predict the order of execution, or how much CPU time will be given.

انت ما بتقولك  
تقول لا OS  
كيف يربط تنفيذهم  
ماالك control  
بس بتقدر تحط  
Priority ل thread  
بناء thread  
دون ضربتك على  
التحكم بال time لا  
Priority

Eng. Asma Abdel Karim  
Computer Engineering Department

Single thread debug multi-thread  
السبب لا

### Multi-threaded Programming (Cont.)



- (a) Threads running on multiple CPUs. *بنفس الوقت بتغذوا.*
- (b) Threads running on a single CPU. *يكون بعمل switch بينهم مش بنفس الوقت بتغذوا.*

Eng. Asma Abdel Karim  
Computer Engineering Department

## Multi-threaded Programming (Cont.)

- Careful attention must be paid to concurrent access and modification of data, to prevent data from becoming out of sync.
- \* With careful design, however, data can be locked, which will prevent read access while write access occurs.
- Multi-threaded programming can be difficult to master, but the rewards that it offers are great.
- \* Networking clients do not need to lock up the GUI if a network connection stalls, and servers can process multiple clients concurrently.
- Additionally, threads may use variables independently, and are not forced to share the same data. → *شاهد*
- \* A thread could, for example, declare its own set of variables that it does not make available to other threads (by marking them as private or protected), thus ensuring that an access conflict does not occur.

لا ضمانه  
data shared  
خلال التنفيذ  
لمنع 2 threads  
ليتم الوصول لـ  
Shared data  
بنفس الوقت

يتخذ Performance  
الحسن من ال design يكون أصعب

ما في داعي  
ال ال GUI يعلق

يعمل ال threads connection

ويكمل يسمع ال User من thread آخر مستوى ال client

## Multi-threading in Java

- Java provides exceptionally good support for creating and running threads and for locking resources to prevent conflicts.
- You can create additional threads to run concurrent tasks in the program.
- Independently* In Java, each task is an instance of the Runnable interface, also called a runnable object.
- A thread is essentially an object that facilitates the execution of a task.

# Multi-threading in Java (Cont.)

Method ال يكون ال implementation بار Super class و أعينه بار sub هاد override أو تكون ال abstract Method بـ abstract class (ما ال implementation) و أنا الطريقة بار sub

```
«interface»
java.lang.Runnable
```

```
java.lang.Thread
```

```
+Thread()
+Thread(task: Runnable)
+start(): void
+isAlive(): boolean
+setPriority(p: int): void
+join(): void
+sleep(millis: long): void
+yield(): void
+interrupt(): void
```

Creates an empty thread.  
Creates a thread for a specified task.  
Starts the thread that causes the run() method to be invoked by the JVM.  
Tests whether the thread is currently running.  
Sets priority p (ranging from 1 to 10) for this thread.  
Waits for this thread to finish.  
Puts a thread to sleep for a specified time in milliseconds.  
Causes a thread to pause temporarily and allow other threads to execute.  
Interrupts this thread.

يخبره override

class عام ما فيه implementation لـ code وال implementation بـ code ال يكون ال extend منه ال class ال ياتنا لـ ال هدف بـ شئ منه إنشاء ال thread بس كل ال thread الي جتعمل جتكون sub منه مش ال هدف منه اني ال object

Eng. Asma Abdel Karim  
Computer Engineering Department

extend thread. في طريقتين لتعريف ال thread هاي أو انك تروح لل class ال ال طريقتين جتعمله thread وتكتب جبه

implements Runnable

بالحاليين بتعمل ال override لل run method. وتكتب فيها ال code ال ال بيك ال يتنفذ لو عند ال thread

## Creating Multi-threaded Applications with the Thread Class

في كذا ال methods الي بتتخدم فيها بال thread

- The `java.lang.Thread` class provides methods to start, suspend, resume, and stop a thread, as well as to control other aspects such as the priority of a thread or the name associated with it.
- The simplest way to use the `Thread` class is to extend it and override the `run()` method, which is invoked when the thread is first started. → هي الي بتستعملها ال thread بتشغل
- By overriding the `run()` method, a thread can be made to perform useful tasks in the background.
- Keep in mind that threads do not start running automatically at creation time. Instead, the `Thread.start()` method must be invoked. If it is not, the thread will not run. → ما بتلعب فيها

Eng. Asma Abdel Karim  
Computer Engineering Department

ال run مو هي ال التي بتشغل ال thread ال التي بتشغله ال method الي اسمها start

لو كتبت run ال start ال thread جتسجل Sequential مش Parallel و هيكت ما بيسر Eng. Asma Abdelkarim

يعني ما بتشغل Sequentially لا بقل الي كان شغال شغال وهنا بتشغل بال background ← in parallel مع ال thread الي شغال (Parent) وهكذا



## Creating Multi-threaded Applications with the Thread Class (Cont.)

هون لسا ما تشغلت thread  
هون ايشغل لما عبت Start.

```

    java.lang.Thread ← CustomThread
    // Custom thread class
    public class CustomThread extends Thread {
    ...
    public CustomThread(...) {
    ...
    }
    // Override the run method in Runnable
    public void run() {
    // Tell system how to perform this task
    ...
    }
    }
    
```

هاد صان class بغير  
اعل من thread  
وبسجلوا  
عادي داخله code زي  
code اي  
لازم نحافظ على الترويسة هيك  
بالتوا وهي أساسية  
نحطها.

```

    // Client class
    public class Client {
    ...
    public void someMethod() {
    ...
    // Create a thread
    CustomThread thread1 = new CustomThread(...);
    // Start a thread
    thread1.start();
    ...
    // Create another thread
    CustomThread thread2 = new CustomThread(...);
    // Start a thread
    thread2.start();
    ...
    }
    }
    
```

هون بيشغل ال thread  
عندي  
\* كل ال thread اللي بعملهم من نفس النوع  
بشاركون نفس ال run method يعني نفس ال code  
بشاركون.

Eng. Asma Abdel Karim  
Computer Engineering Department

## Creating Multi-threaded Applications with the Thread Class (Example)

mythread ←

```

    public class ExtendThreadDemo extends java.lang.Thread {
    int threadNumber;
    public ExtendThreadDemo ( int num ){
    // Assign to member variable
    threadNumber = num;
    }
    // Run method is executed when thread first started
    public void run(){
    System.out.println ("I am thread number " + threadNumber);
    try{
    // Sleep for five thousand milliseconds (5 secs), to simulate work being done
    Thread.sleep(5000);
    } catch (InterruptedException ie) {
    System.out.println (threadNumber + " is finished!");
    }
    }
    }
    
```

هيك تالته يورث  
مغات ال thread  
من ال thread class  
وال methods  
Static Method → if I want to use delay  
بسط على ال checked  
checked exception (you must handle it.)

ال ال thread  
التي اتسببت منه  
هاي ال Method  
حيث ان ال  
checked exception  
تسبب ال  
مشكلتها انه يتدخل  
تستهلك ال cpu time

Eng. Asma Abdel Karim  
Computer Engineering Department

# Creating Multi-threaded Applications with the Thread Class (Example)

```
// Main method to create and start threads
public static void main(String args[]){
    System.out.println ("Creating thread 1");
    // Create first thread instance
    Thread t1 = new ExtendThreadDemo(1);
    System.out.println ("Creating thread 2");
    // Create second thread instance
    Thread t2 = new ExtendThreadDemo(2);
    // Start both threads
    t1.start(); t2.start();
}
```

← الأغب t, بطبع  
 جملة بعين بنام 5  
 ثواني بنفس هالوقت  
 t2 بتشغل ويطبع أول  
 جملة وبنام 5 ثواني  
 بعين الاثنين  
 بيقولوا ويطبعوا  
 ثاني جملتين

Polimorfism  
 هو انما له معنى لانه  
 انا بعمل thread  
 من انواع مختلفة عندي  
 بس نوع ، فكنت بقدر  
 بدل thread اكتب ExtendThreadDemo

لو نليت هيك ما يكون شغلت → t1.run / t2.run  
 ال thread قال output يكون واحد معروف لانه sequential run  
 1- أول جملة ل t1  
 2- بنام 5 ثواني

ترتيب أول جملتين  
 بخلاف حساب ال OS  
 وكذلك ثاني  
 جملتين ، لكن  
 أول 2 أكيد جملتين  
 قبل أخذ 2 لانه  
 في Sleep بينهم  
 5 seconds

- 3- ثاني جملة ل t1
- 4- أول جملة ل t2
- 5- بنام 5 ثواني
- 6- ثاني جملة ل t2

# Creating Multi-threaded Applications with the Thread Class (Notes)

- The run() method is not invoked when the thread was created, only when the thread is started by invoking the start() method.
- \* - You can create threads in advance, and start them only when needed. →
- Remember that the thread object only represents a thread—threads are in fact provided by the operating system itself.
- \* - When the start() method of a thread is called, it sends a request to launch a separate thread, which will call the run() method.
- \* - The main application does not call the run() method directly. Instead, it calls start() to perform this operation. If your application calls run() directly, it won't be running as a separate thread.

← هي جملتين  
 بتنادي ال  
 run method  
 جوا ال thread

هو بفهم ال OS انه هاد  
 thread ، فواي مسقطية ال Start

\* ال email App مثال على استخدام ال Daemon thread ،  
 لو أنا كنت بكتب الرسالة وبعد attach وهيك وقريت أبطل ايجت  
 ال email فيسكوت ال main thread ← thread الكتابة و attach // thread  
 بطل الهم معنى . استخدم " setDemon " .

by default ← ال thread يكون user لو بي أنظير ال Demon

## Creating Multi-threaded Applications with the Thread Class (Notes)

- The main method terminates once the two threads are started.
- There is no pause or sleep command issued in the main thread—yet the application doesn't terminate. It keeps on going until the two threads have finished their work and leave their run() method.
- When a normal thread (also referred to as a user thread) is created, it is expected that it will complete its work and not shut down prematurely.
- The Java Virtual Machine (JVM) will not terminate until all user threads have finished, or until a call is made to the System.exit() method, which terminates the JVM abruptly.
- Sometimes, however, threads are only useful when other threads are running (such as the actual application, which will eventually terminate when the user is finished with it).
- We call these types of threads daemon threads, as opposed to user threads. If only daemon threads are running, the JVM will automatically terminate.

انتهاء ال main thread  
 التي تشغيل ال thread  
 هل معناه انهم  
 خلصوا فعلا ؟  
 ما حطينا Rise  
 أو sleep لنسكوتهم  
 فال thread التي ما  
 خلصوا ما الهم علاقة  
 ميدي تشغيل لحد ما  
 t1 و t2 يخلصوا  
 ال run تاوم .  
 مع انه ال thread  
 التي خلصوا خلص  
 بيهم ال Children

مفيد لما يكون ال thread التي انصفت  
 بتعتمد تشغيل ال main thread التي علقت  
 فلو كل ال تشغيل ال main thread  
 يكون ← مجرد ما ال creator تاوم  
 بخلص كل الباقي  
 ينتهوا .

Eng. Asma Abdel Karim  
 Computer Engineering Department

19 Children ما راخوم (هدول اسوم ال User thread) : بخلصوا تنفيذ تنقوا ال thread الأب التي أنشأهم خلص .

## Daemon Threads

- The following is a modification to the previous main method such that t1 and t2 are specified as daemon methods.

```
public static void main(String args[]){
    System.out.println("Creating thread 1");
    // Create first thread instance
    Thread t1 = new ExtendThreadDemo(1);
    System.out.println("Creating thread 2");
    // Create second thread instance
    Thread t2 = new ExtendThreadDemo(2);
    // Make both threads daemon threads
    t1.setDaemon(true); t2.setDaemon(true);
    // Start both threads
    t1.start(); t2.start();
    try{
        // Sleep for one second, to allow threads time to display first message
        Thread.sleep(1000);
    }
    catch (InterruptedException ie) {}
}
```

لازم اعلوا قبل  
 ما أشغله .  
 لأنه لو علته  
 بعد الالفاني  
 يكون  
 انشغل  
 حالة  
 ال User thread  
 ← user ← False  
 ما جيلتقوا يطبعوا ثاني جملة  
 لأنه أصلا هم عاملين

Eng. Asma Abdel Karim  
 Computer Engineering Department

20 Sleep لمدة 5 ثواني . كطية مجال 1 sec  
 انه يطبع اول جملة  
 أما الجملتين التانيتين ما جيلتقوا  
 يطبعوها لأنه ال main thread  
 يكون خلص بعد الثانية .

## Daemon Threads (Notes)

- The first change makes both t1 and t2 daemon threads, by calling the `setDaemon(boolean)` method.
- If you need to change the state of a thread to either a daemon or a user thread, this must be done before the thread is started—its state cannot be changed once the thread is running.
- The second change introduces a slight pause, to allow the daemon threads time to display their first message.
  - \*— When you recompile and run this example, you'll notice that the threads do not complete their work and display their final message. This is because there are no more user threads active once the main method finishes.
- The primary thread is always a user thread, never a daemon thread.
 

↓  
main thread



Eng. Asma Abdel Karim  
Computer Engineering Department

21

21

## Creating Multi-threaded Applications with the Runnable Interface

- While extending the *Thread* class is one way to create a multi-threaded application, it isn't always the best way.
- Remember, Java supports only single inheritance, unlike languages such as C++, which supports multiple inheritance.
- This means that if a class extends the *java.lang.Thread* class, it cannot extend any other class.
- A better way is often to implement the *java.lang.Runnable* interface.

بكون خسرنا  
فحسبنا لا  
لانها بقدر احد  
أكثر من  
دايا مسرجلي  
وحدة بكون و  
أنا صرقتوا عاد  
Thread  
فما بقدر أعمل  
class ل extend  
تأخا لو محتاجة.



Eng. Asma Abdel Karim  
Computer Engineering Department

22

22

## Creating Multi-threaded Applications with the Runnable Interface (Cont.)

- The Runnable interface defines a single method, *run()*, that must be implemented.
- Classes implement this interface to show that they are capable of being run as a separate thread of execution. → *أعطيت صيغة: إذا بقدرنا عمل run*
- The precise signature for the run method is as follows: *concurrently*

**public void run ()**



## Creating Multi-threaded Applications with the Runnable Interface (Cont.)

```

// Custom task class
public class TaskClass implements Runnable {
    ...
    public TaskClass(...) {
        ...
    }
    // Implement the run method in Runnable
    public void run() {
        // Tell system how to run custom thread
    }
}

// Client class
public class Client {
    ...
    public void someMethod() {
        ...
        // Create an instance of TaskClass
        TaskClass task = new TaskClass(...);
        // Create a thread
        Thread thread = new Thread(task);
        // Start a thread
        thread.start();
    }
}
    
```

*object Runnable يدبر  
Start thread لما لا thread  
أخبرنا نوع thread  
هناك إنشاء thread  
عن طريق الobject Runnable  
كيفية بغير عمل Start*

*object  
مخزن صفة  
Runnable  
بين عنوان عمل  
start لا شيء  
لا يمكن thread  
بالنهاية.*



← ما بتكون thread لازم انا اتمق constructor ل thread لينزيد  
 Runnable بي تعطيني صفة ال run in parallel ما بتعطيني  
 ال methods Start و غيره ، هقول ال Methods يكونا بال  
 Thread object

## Creating Multi-threaded Applications with the Runnable Interface (Cont.)

- The Runnable interface doesn't define any other methods, or provide any thread-specific functionality.
- Its sole purpose is to identify classes capable of running as threads.
- When an object implementing the Runnable interface is passed to the constructor of a thread, and the thread's start() method is invoked, the run() method will be called by the newly created thread.
- When the run() method terminates, the thread stops executing.

## Creating Multi-threaded Applications with the Runnable Interface (Example)

```
public class RunnableThreadDemo implements java.lang.Runnable{
    public void run(){
        System.out.println ("I am an instance of the java.lang.Runnable interface");
    }
    public static void main(String args[]){
        System.out.println ("Creating runnable object");
        // Create runnable object
        Runnable run = new RunnableThreadDemo();
        // Create a thread, and pass the runnable object
        System.out.println ("Creating first thread");
        Thread t1 = new Thread (run);
        // Create a second thread, and pass the runnable object
        System.out.println ("Creating second thread");
        Thread t2 = new Thread (run);
        // Start both threads
        System.out.println ("Starting both threads");
        t1.start(); t2.start();
    }
}
```

لو بيدل Runnable كتب اسم كلاس  
 علي ما بتفرق .  
 \* بقدر اعرق runnable  
 واحد واعمله اكثر من thread  
 عن نفس النوع (shared data/code)

## Creating Multi-threaded Applications with the Runnable Interface (Example-Notes)

- When the example is compiled and run, two threads can be seen printing a message to the console.
- What is very different about this program, and the previous one, is that only one **Runnable** object was created, but two different threads ran it.
- Although there was no shared data in this example, in more complex systems, threads must share access to resources, to prevent modification while a resource is being accessed. This is achieved by synchronizing access to resources

تأكد لمتنوعا يدير في  
• race condition

Eng. Asma Abdel Karim  
Computer Engineering Department

27

27

## Advantages of Using the Runnable Interface over Extending the Thread Class

1. As mentioned previously, an object is free to inherit from a different class. → يقتر العمل extend ل class لو بدى
2. The same Runnable object can be passed to more than one thread, so several concurrent threads can be using the same code and acting on the same data.
  - \* - Though this use is not always advised, it can make sense in certain circumstances, providing that due care is taken to prevent conflicts over data access. القصد لازم تكلمنا منتهيه عشان ما يفسد اخطا
3. Carefully designed applications can minimize overhead, as creating a new Thread instance requires valuable memory and CPU time.
  - \* - A Runnable instance, on the other hand, doesn't incur the same burden of a thread, and can still be passed to a thread at a later point in time to be reused and run again if necessary. ينقل ال overhead

لانص بالطريقة الاولى لان كل thread يكون له  
ال data/code الخاص فيه

يعمل Runnable  
object واحد ويجعل  
منه اكثر من thread  
وهيك بسهولة  
ينقل ال Code /  
data يكونوا  
Shared بين  
ال thread

Eng. Asma Abdel Karim  
Computer Engineering Department

28

28

## Controlling Threads

نظري one thread يقطف  
ال Sleeping  
other thread =

### ① Interrupting a Thread

- Observant readers may have noticed that whenever a call to the **Thread.sleep(int)** method was made in earlier examples, an exception handler was used.   
 *بناديها باسم ال class علي*
- This is because the **sleep** method puts a thread to sleep for a long period of time, during which it is generally unable to rouse itself.   
 *ال thread ما يفيق إلا لتخلص التناهي أو thread تفيقه*
- However, if a thread must be awakened earlier, interrupting a thread will awaken it; this is achieved by invoking the **interrupt()** method.
- Of course, this requires another thread to maintain a reference to the sleeping thread.   
 *لأنه لازم ال thread التاني يكون عنده ال reference ال thread النائم*

*عشان يقدر يفيقه قبل صوته*

## Controlling Threads

### Interrupting a Thread (Example)

```
public class SleepyHead extends Thread{
    // Run method is executed when thread first started
    public void run(){
        System.out.println ("I feel sleepy. Wake me in eight hours");
        try{
            // Sleep for eight hours
            Thread.sleep( 1000 * 60 * 60 * 8 );
            System.out.println ("That was a nice nap");
        }
        catch (InterruptedException ie){
            System.err.println ("Just five more minutes....");
        }
    }
}
```

*انا خلص نوم 8 ساعات  
بفيقا من حاله وينغد هاي  
أما لو ال thread ثانية قطعت قبل  
ال 8 حيطع هاي  
ما حيطع ال  
"that was a nice nap"*



## Controlling Threads Interrupting a Thread (Example-Cont.)

```
// Main method to create and start threads
public static void main(String args[]) throws java.io.IOException{
    // Create a 'sleepy' thread
    Thread sleepy = new SleepyHead();
    // Start thread sleeping
    sleepy.start();
    // Prompt user and wait for input
    System.out.println ("Press enter to interrupt the thread");
    System.in.read();
    // Interrupt the thread
    sleepy.interrupt();
}
```

يقدر بيل thread اكتب SleepyHead  
 blocking until the user enter something  
 لو دخل اشئ ال user حينئذى Sleepy.interrupt  
 فتح تتسالى ال interrupt  
 وبكفى خالص ال main و خالص ال Sleepy

\* ال main قدر يعمل interrupt لأنه يعرف ال reference  
 ناع ال thread ، لو ال user ما كبس ال thread بنام ال 8 كاملين

## Controlling Threads Stopping, Suspending and Resuming a Thread

- The Thread class also contains the stop(), suspend(), and resume() methods.
- As of Java 2, these methods were deprecated (or outdated) because they are known to be inherently unsafe.

thread ينوي تنفيذ  
 thread فاف كنه ال ref  
 لـ ينوي تنفيذ thread تاني مؤقت در ما يايي  
 مشكلة هصول ال method كلوا يعملوا مشاكل  
 ال execution ويسبوا dead locks (حالة اعتماد معلقة بين ال thread)  
 deprecated بطولوا بيستخدموا بسوا حاولت تستشيرهم  
 على بتقرر بس لا يوضح فيهم

بهمال  
 فعاليتها  
 الزمن ت

\* كل هصول ال method ال thread ال class والجملين مش مخطوطة اسمائهم  
 ال am

Static ← **Controlling Threads** من العمليات التي  
المعكثه بقررها  
عنا زي الستا  
مثلا زي الباقيين

**⑤ Yielding CPU Time**

- Sometimes a thread might be waiting for an event to occur, or may be entering a section of code where releasing CPU time to another thread will improve either system performance or the user experience.
- For example:
  - \* - After performing a calculation that should be displayed to the user and before starting another one.
  - \* - While waiting for data to become available from an *InputStream*, a thread might yield CPU time instead of going to sleep. → أعطي موجة للداتا أتومل
- In this situation, the static *yield()* method can be used instead of the *sleep()* method.

شبه بعض شوي بستخدمه اما بدنا  
ال thread ماشوي اشئ. الفرق اننا yield بتخلي  
ال thread ما يستولك من ال CPU time

Eng. Asma Abdel Karim  
Computer Engineering Department

**Controlling Threads**  
**Yielding CPU Time (Cont.)**

- For example, for the currently running thread to yield CPU time, the following method could be invoked:

`Thread.yield();` ← ال thread نفسه هو اللي يستدعينا  
ما يجيب thread  
ثاني يستدعينا  
بلوا.

- This is a static method that affects the currently running thread only—an application cannot yield the time of a specific thread.

\* الوقت اللي بتعمل فيه yield مش انت بتحدده  
ال OS بحدده هنا شوي هو كويس بس عارفين

Eng. Asma Abdel Karim  
Computer Engineering Department

## Controlling Threads Waiting Until a Thread is Dead

- Sometimes it is necessary to wait until a thread has finished its task.
- \* - For example, to retrieve the results of the task by invoking a method, or reading a member variable.
- To determine if a thread has died (i.e., if the run() method has finished), the isAlive() method, which returns a boolean value, can be invoked.
- But continually checking the value returned by this method (known as polling), and then sleeping or yielding, is a very inefficient use of CPU time.

thread به يعرف  
thread يعرف  
معين انتهى  
لأسباب معينة في  
الـ execution.

ترجع true  
لو لسا شغال.

ترجع هذا الـ thread هو alive ولا لا.  
منش طريقة  
مناسبة كثير فيستخدم بعد اسأل بواجي الـ method لحد ما عرف مات أو لا.

35

Ex: while (tL.isAlive()) {  
}   
— حيثل يشيك وايما فلو بيدي احد فرق بسيط بينا  
كل تشيئة بعد sleep أو yield بال body

Ex: while (tL.isAlive()) {  
}

## Controlling Threads Waiting Until a Thread is Dead (Cont.)

- A much better way is to use the join() method, which waits for a thread to die.
- There is also an overloaded version of this method, which takes as a parameter a long value. This version waits for a thread death or the specified number of milliseconds, whichever comes first.

method blocking  
حيط معلق فيها  
التنفيذ لحد ما الـ  
thread الي انت  
منادى عليه الـ join  
ينتهي فيتكمل  
تنفيذ البرنامج  
بغير الـ thread  
واقف بـ queue

\* طريقة أفضل ما يستهلك  
cpu time  
\* صار الـ join تحديث الـ الذي بتقرر تحل بعد ثواني  
بالاقواسا > انا كلص خلال الزمان  
بكمل لوما كلص برضه بكمل لأنه خلاصت الـ مرة  
المسوحة للانتظار

36

لحد ما الـ OS يخبرك انه هلاك الـ thread مات  
فانت بتكمل تنفيذ.

# Controlling Threads Waiting Until a Thread is Dead (Example)

```

public class WaitForDeath extends Thread{
    // Run method is executed when thread first started
    public void run(){
        System.out.println ("This thread feels a little ill....");
        // Sleep for five seconds
        try{
            Thread.sleep(5000);
        }
        catch (InterruptedException ie) {}
    }
}

```

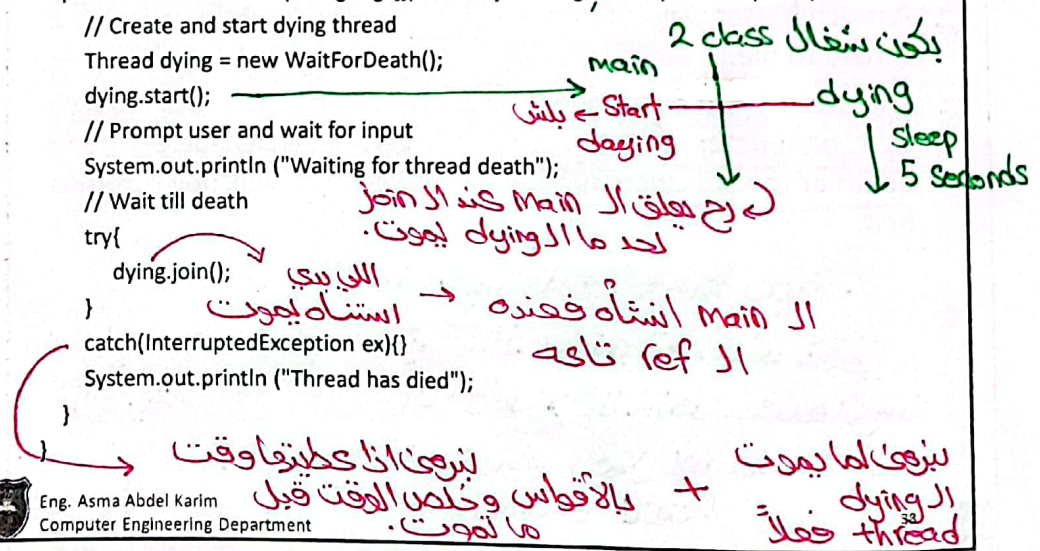
كلمة بقلك ال thread  
مريضاً وكينيمه  
5 ثواني  
بعدين حيكون  
مات لانه التوي اتغيزده.

# Controlling Threads Waiting Until a Thread is Dead (Example-Cont.)

```

// Main method to create and start threads
public static void main(String args[]) throws java.lang.InterruptedException{
    // Create and start dying thread
    Thread dying = new WaitForDeath();
    dying.start();
    // Prompt user and wait for input
    System.out.println ("Waiting for thread death");
    // Wait till death
    try{
        dying.join();
    }
    catch(InterruptedException ex){}
    System.out.println ("Thread has died");
}

```



ترتيبهم ما يفرق

38  
 This thread feels a little ill output ال  
 ← waiting for thread death  
 Thread has died →

هلي الكيد  
 آخر وحدة.

حكيما انه ما لنا Control على OS لينخذ أي thread أول. بس بقدر أوكي Priority لبعض ال threads عشوائيا ال OS يفوز معهم.

### Threads Priorities

- Java assigns every thread a priority.
- By default, a thread inherits the priority of the thread that spawned it. → ال thread ياخذ ال Priority تابعاً من ال thread اللى شغلها
- You can increase or decrease the priority of any thread by using the setPriority method, and you can get the thread's priority by using the getPriority method.
- Priorities are numbers ranging from 1 to 10. → الرقم الأعلى يمثل ال Priority الأعلى.
- The Thread class has the int constants MIN\_PRIORITY, NORM\_PRIORITY, and MAX\_PRIORITY, representing 1, 5, and 10, respectively. → constants → ال Priority اللى الـ static also.
- The priority of the main thread is Thread.NORM\_PRIORITY.

```
Ex: t1.setPriority(MAX_PRIORITY);
    ≡ t1.setPriority(10);
```

39 ال main ال Priority تابعاً دائماً اللى (5)

### Threads Priorities (Cont.)

- The JVM always picks the currently runnable thread with the highest priority. → ال Priority وقت أكثر لي ال أعلى ال Priority
- A lower-priority thread can run only when no higher-priority threads are running. → مش طالما ال thread ال Priority ما يشغل ما يشغل
- If all runnable threads have equal priorities, each is assigned an equal portion of the CPU time in a circular queue.
- \* - This is called round-robin scheduling.

عشان ما يجيب Starvation. انه الأقل ما يفعل دائماً.

كده السلايب الكلام هو دقيقاً فينوع ال Scheduling لأنه هذه الأمور حسب نوع ال Scheduling ياخذ ال OS

لا يمكن thread يعمل Shared data ممكن يترتب عليه موضوع  
 خطر ( conflict on the data ) أكثر من thread يعمل access  
 نفس ال Variable بنفس الوقت.

12/21/20

## Thread Synchronization

- An important consideration when designing multi-threaded applications is conflict over access to data.
- If two threads are fighting for the same resource, and a mechanism to resolve access conflicts is not put into place, the integrity of the application is at stake. *لازم ما تتناقض وحدة البيانات*
- Built into the Java language are two mechanisms for preventing concurrent access to resources:
  - ① - Method-level synchronization and,
  - ② - Block-level synchronization.



Eng. Asma Abdel Karim  
 Computer Engineering Department

41

41

حتمين 2 threads لازم يتغلبوا Methods نفس ال object بنفس الوقت.  
 لو objects مختلفة عادي ما في مشكلة.

## Method-Level Synchronization

- Method-level synchronization prevents two threads from executing methods on an object at the same time.
- Methods that must be "thread-safe" are marked as synchronized.
- When a synchronized method of an object is invoked, a thread takes out an object lock, or monitor.
  - \* - If another thread attempts to execute any synchronized method, it finds that it is locked, and enters a state of suspension until the lock on the object monitor is released.
- If several threads attempt to execute a method on a locked object, a queue of suspended threads will form.
  - \* - When the thread that instituted the lock returns from the method, only one of the queued threads may access the object—the release of a monitor does not allow more than one object to take out a new monitor.
- One should note, however, that if a method is not synchronized and is executed while the object is locked, the thread will not block and the method can be run.

مراح يفوت  
 thread ثاني  
 يشغل نفس  
 هذا ال object.  
 ينص زي ال lock  
 فقط لا  
 Synchronized  
 methods  
 الباقي عادي  
 ما في مشاكل.



Eng. Asma Abdel Karim  
 Computer Engineering Department

42

42

بجبر الي بخلص بيتترك ال lock بيأخذه  
 لوجه واحد بس وهكذا.

حتفا لو كان في Method Synchronized  
 بتغلق نفس ال object.

ما تسمح التنفيذ فقط لو  
 كانت Synchronized  
 ونفس ال object.

Eng. Asma Abdelkarim

\* بحريش تفعل كذا ال Method تكون Synchronized هيكله كذا ال رجعت  
 تسغل Serial. انت تعرف بس اللي بتشاركو ويكتبوا ويقرأوا  
 نفس ال data.

## Method-Level Synchronization (Cont.)

- The synchronized keyword is used to indicate that a method should be protected by a monitor. *جا ال صهاي الكلفي*
- Every method that could possibly be affected by concurrent access should be marked as synchronized. This keyword should be used sparingly, however, as it has a performance drawback.

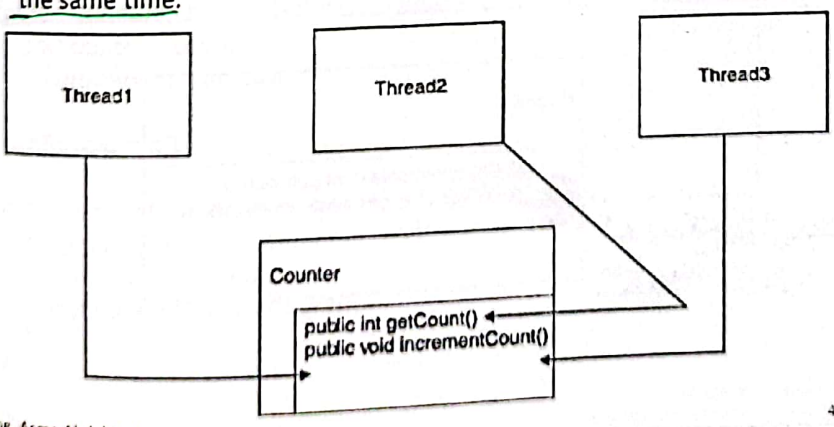
```
public class SomeClass{
    public synchronized void changeData(...){
        .....
    }
    public synchronized Object getData (...){
        .....
    }
}
```

بجدر استخدام  
 أقل استخدام  
 بس للضرورة

بعض ما ينقطع  
 بنفس الوقت

## Method-Level Synchronization (Cont.)

- Suppose we have a counter that can both be incremented and display a value.
- If the methods that provides access to the counter isn't thread-safe, and takes some time to complete, then two or more threads could access it at the same time.



Thread 3, 1  
 ممكن مع بعض  
 تنفذوا فيلكوا  
 ناتج ال count  
 غلط. فزيك  
 مافي ضمانة  
 لصحة البيانات.

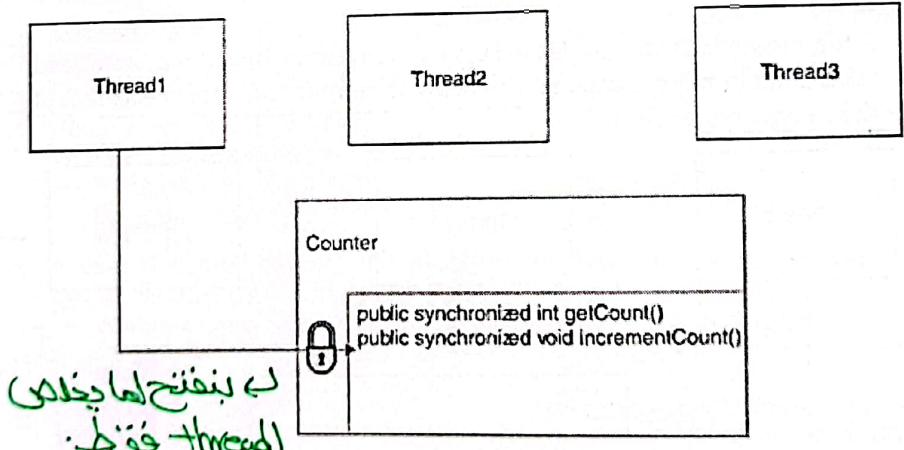
## Method-Level Synchronization (Cont.)

- The solution is to make the counter thread-safe, by synchronizing each method that performs a read or write operation.
- If a synchronized method is used, only one thread can update the value at any given moment.
  - \* - The thread that first invokes a synchronized method locks the object's monitor, which is released only when that method terminates.
  - \* - No other thread can access any synchronized method of the counter object.
- This restriction applies only to individual counter instances, and not the Counter class itself. ↴

object ال Synchronization يتم فقط نفس ال object  
 Synchronized Methods اذا جات انا في نفس ال object  
 ثاني ما انا علاقة بال موضوع  
 45

فانا هي مش في مستوى ال class و مستوى ال object

## Method-Level Synchronization (Cont.)





## Method-Level Synchronization (Example)

```

public class Counter{
    private int countValue;
    public Counter(){
        countValue = 0;
    }
    public Counter(int start){
        countValue = start;
    }
}

// Synchronized method to increase counter
public synchronized void increaseCount(){
    int count = countValue;
    try{
        Thread.sleep(5);
    }
    catch (InterruptedException ie) {}
    count = count + 1;
    countValue = count;
}

// Synchronized method to return counter value
public synchronized int getCount(){
    return countValue;
}
}

```

عملية  
فيون  
فقط  
لنظي  
الوقت لا  
Method



Eng. Asma Abdel Karim  
Computer Engineering Department

47

47

## Method-Level Synchronization (Example-Cont.)

```

public class CountingThread implements Runnable{
    Counter myCounter;
    int countAmount;
    // Construct a counting thread to use the specified counter
    public CountingThread (Counter counter, int amount){
        myCounter = counter;
        countAmount = amount;
    }
    public void run()
    {
        // Increase the counter the specified number of times
        for (int i = 1; i <= countAmount; i++){
            // Increase the counter
            myCounter.increaseCount();
        }
    }
}

```

عن طريقنا  
الشايف  
أسهل بين  
الThreads

المتغير  
المتغير  
المتغير  
المتغير



Eng. Asma Abdel Karim  
Computer Engineering Department

48

48

## Method-Level Synchronization (Example-Cont.)

```
public static void main(String args[]) throws Exception{
    // Create a new, thread-safe counter
    Counter c = new Counter();
    // Our runnable instance will increase the counter
    // ten times, for each thread that runs it
    Runnable runner = new CountingThread( c, 10 );
    System.out.println ("Starting counting threads");
    Thread t1 = new Thread(runner);
    Thread t2 = new Thread(runner);
    Thread t3 = new Thread(runner);
    t1.start(); t2.start(); t3.start();
    // Wait for all three threads to finish
    t1.join(); t2.join(); t3.join();
    System.out.println ("Counter value is " + c.getCount() );
}
```

بعض اعمل منه  
Thread objects  
وكلهم يكونوا  
shared  
لا Counter  
وال  
countAmount

كل واحد حيعمل increase بقيمة واحدة بالزبط  
وما في أكثر من حد حيعمل increase  
بنفس الوقت.

بالنواية قيمة ال counter حتكون 30

## Block-Level Synchronization

- Method-level synchronization is an effective means of preventing concurrent access to resources.
- But what if the resource has not been designed as thread-safe, and is a preexisting class that the developer cannot modify?
  - \*— Such as a class in the Java API, or a third-party library.
- Block-level synchronization, in this case, is the best option.

بشغل مستوى ال block  
اللي كاتبها ال methods  
فما يقدر اتحكم  
بالسynchroniz  
أو لو كان بي  
جزع من ال  
method  
يكون سynchroniz  
مش كله.

\* أي شيء بين { } يسمى block . وعاى زفول block بجزء ال code

## Block-Level Synchronization (Cont.)

- Block-level synchronization uses the synchronized keyword, but instead of placing a lock around particular methods, a lock is placed around blocks of code.
- A block of code is synchronized against a particular object, and any thread attempting to enter that block of code is locked out, until the monitor for the specified object is released.
- The following code snippet shows the syntax for a synchronized block:

```
synchronized (Object o){
    .....
}
```

ال object  
اللى بيستخدم  
Synchronization  
عليه



Eng. Asma Abdel Karim  
Computer Engineering Department

51

51

## Block-Level Synchronization (Cont.)

- Block-level synchronization locks against a particular object.
- This means that multiple blocks can protect access to the same object, so block-level synchronization can be applied in thread code wherever an object is accessed or modified.

\* نفس ال synchronize ال Method - level  
بس عاى بجزء ال block



Eng. Asma Abdel Karim  
Computer Engineering Department

52

52

Asma Abdelkarim

## Block-Level Synchronization (Example)

```
public class SynchBlock implements Runnable{
    StringBuffer buffer;
    int counter;
    public SynchBlock(){
        buffer = new StringBuffer();
        counter = 1;
    }
}
```

← نستخدمه  
لما بدنا نعدل  
كثير على String  
لا بد جيب بنفس  
المكان ما يخلط  
array حديقة  
للتفديزي  
ال String



Eng. Asma Abdel Karim  
Computer Engineering Department

53

53

## Block-Level Synchronization (Example)

```
public void run(){
    synchronized (buffer){
        System.out.print ("Starting synchronized block ");
        int tempVariable = counter++;
        // Create message to add to buffer, including linefeed
        String message = "Count value is : " + tempVariable +
            System.getProperty("line.separator");
        try{
            Thread.sleep(100);
        }
        catch (InterruptedException ie) {}
        buffer.append (message);
        System.out.println ("... ending synchronized block");
    }
}
```

← وهادي اكثر من 2  
ما بي يسووها

← إضافة  
line  
طاليني  
هيك  
لتخمين  
الوقت

← هادي الي ما بي  
أكثر من 2  
يسووها



Eng. Asma Abdel Karim  
Computer Engineering Department

54

54

Eng. Asma Abdelkarim

## Block-Level Synchronization (Example)

```
public static void main(String args[]) throws Exception{
```

```
// Create a new runnable instance
```

```
SynchBlock block = new SynchBlock();
```

```
Thread t1 = new Thread (block);
```

```
Thread t2 = new Thread (block);
```

```
Thread t3 = new Thread (block);
```

```
Thread t4 = new Thread (block);
```

```
t1.start(); t2.start(); t3.start(); t4.start();
```

```
// Wait for all these threads to finish
```

```
t1.join(); t2.join(); t3.join(); t4.join();
```

```
System.out.println (block.buffer);
```

```
}
```

StringBuffer  
بطريقة  
buffer

Runnable object  
Sharing buffer and counter

حيزنا في الـ buffer  
4 string



Eng. Asma Abdel Karim  
Computer Engineering Department

55

55

Synchronized String Buffer \*  
Synchronized String Builder \*

## Inter-Thread Communication

- A design that requires no communication between threads lends itself to a far simpler implementation.
- However, sometimes it is necessary for threads to communicate with each other.
- Often, the type of communication will be fairly simple, such as reading or modifying a public member variable, or invoking an object method.
- Two good options for communication are:
  - 1 - Communication pipes and,
  - 2 - The wait()/notify() methods, which allow one thread to notify a waiting thread of an event.

مات يحتاج  
يكون في تواصل  
بين الـ thread  
وهنا بصعب  
كلينا التنفيذ  
شوي

actual communication

من زبي الطرق

اللي اخذناها قبل



Eng. Asma Abdel Karim  
Computer Engineering Department

56

56

thread يستعمل thread لتوصيل شئ معين  
أوهيل

### Communication Pipes between Threads

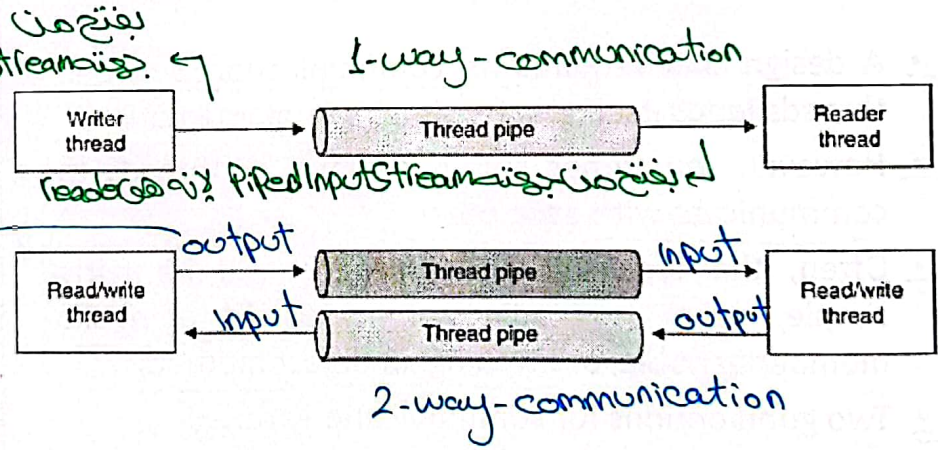
- Like multi-process communication, which uses pipes to send data from one process to another, threads can also send data directly from one thread to another.
- This is achieved by using special types of input and output streams, which are linked together.
  - \*- By passing either end of the pipe to another thread, that thread may listen to, or speak to, another thread.
  - \*- In fact, there's no restriction preventing two pipes from being used—a thread could even have two-way communication with another.

في مزوع reader / writer كمان هاي بتطلب تفتح هويتين مناظروا حد

Piped input و Piped output مع بعض ، تشبك مع اعملهم link او الكس

Eng. Asma Abdel Karim  
Computer Engineering Department

### Communication Pipes between Threads (Cont.)



بفتح من PipedOutputStream لانه هو writer

بفتح من PipedInputStream لانه هو reader

واحد

منع بمرقده reference الثاني ليحسب لينعم توصل و كذاك بار

1-way-comm

\* لازم واحد من ال thread قادر يمرق ال reference تلو للطرف الثاني عشان يبين بينهم توصل .

Eng. Asma Abdel Karim  
Computer Engineering Department

## Communication Pipes between Threads (Example)

```
import java.io.*;
public class PipeDemo extends Thread{
    PipedOutputStream output;
    // Create an instance of the PipeDemo class
    public PipeDemo(PipedOutputStream out){
        // Copy to local member variable
        output = out;
    }
}
```



Eng. Asma Abdel Karim  
Computer Engineering Department

59

59

## Communication Pipes between Threads (Example-Cont.)

```
public static void main (String args[]){
    try{
        // Create a pipe for writing
        PipedOutputStream out = new PipedOutputStream();
        // Create a pipe for reading, and connect it to output pipe
        PipedInputStream pin = new PipedInputStream(out);
        // Create a new pipe demo thread, to write to our pipe
        PipeDemo pipedemo = new PipeDemo(out);
        // Start the thread
        pipedemo.start();
        // Read thread data,
        int input = pin.read();
    }
}
```

مبليا على not connected

مرقلا ال output تاغلا

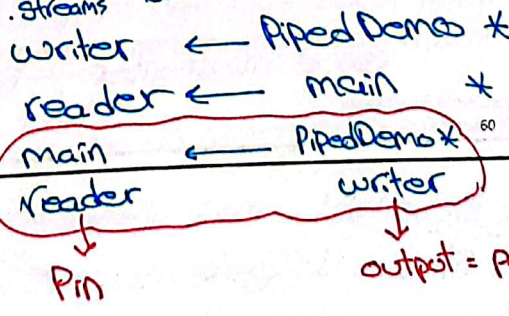
انشا ال thread

شغلوا

قدر يتكلم ب صوا ، بتقدر تخط . streams level

تجسنتا PipeDemo ليكتب data ميثان يقرأها

الهدف يتلبي ال Pin للاستخدام الخاص لانه هو ال reader وال Post بده يرمقه لل thread الي على



Eng. Asma Abdel Karim  
Computer Engineering Department

60

60

## Communication Pipes between Threads (Example-Cont.)

```
// Terminate when end of stream reached
while (input != -1){
    // Print message
    System.out.print ( (char) input);
    // Read next byte
    input = pin.read();
}
}
catch (Exception e){
    System.err.println ("Pipe error " + e);
}
}
```

*end of stream* (pointing to `input != -1`)

*cast* (pointing to `(char) input`)



Eng. Asma Abdel Karim  
Computer Engineering Department

61

61

## Communication Pipes between Threads (Example-Cont.)

```
public void run(){
    try{
        // Create a printstream for convenient writing
        PrintStream p = new PrintStream( output );
        // Print message
        p.println ("Hello from another thread, via pipes!");
        // Close the stream
        p.close();
    }
    catch (Exception e){
        // no code req'd
    }
}
}
```

*must write because pipedDemo is writer.* (pointing to `run()`)

*= pout* (pointing to `PrintStream( output )`)

*تنظف  
جوجل ال main* (pointing to `p.close()`)



Eng. Asma Abdel Karim  
Computer Engineering Department

62

62

*\* يتقدر تفصل PipedReader / PipedWriter لو بي  
الحكي يكون text*

*يتقدر تخلي 2 thread يحدوا سا وال Main سا بنوا هي الي  
لتنش ال Streams ويتشرك مع بعض وانتر كل reference Stream  
لواحد من ال threads فيجب بحيروا مشركين  
مع بعض*

Eng. Asma Abdelkarim



## Notifying a Waiting Thread of an Event

- A common requirement in multi-threaded programming is that one thread cannot proceed until the completion of a task by another thread.
- \* - Sometimes a thread will be producing information or using resources. Other times, the order of execution is important, and a task cannot take place before another has completed.
- While it is possible for one thread to wait until another has died (thus indicating that the work was completed) by using the Thread.join() method, what if a thread performs an ongoing task and never terminates?

task ل wait  
 صينية مش  
 لخدمة ال thread  
 يموت  
 زي قبل ما  
 حينا.



## Notifying a Waiting Thread of an Event (Cont.)

- The solution is to notify other threads that a task has been completed.
- \* - Threads wait until they are notified, and notification can be a repeated process (with several cycles of waiting and notifying).
- This allows threads to synchronize their actions and communicate that a critical event has occurred, without requiring the extra complexity of pipe-based communication or invoking methods.
- \* - Sometimes a thread may not even know exactly which threads are waiting for it to complete, so a special type of notification is used.

بعضنا ال thread اللي بيشغره ليطلم task

اللي صناديق ال wait

كشمان ال Pipes  
 → complex  
 فيستخد الطريقة

مش ضروري ال thread يعرف



• مين ال thread اللي بيشغره  
 • فلذلك هو flexible

\* كل Object فيه القرة انه يخلي threads يستريح وينه القرة ينادي notify ليعمل released ليعمل threads من ال waiting

### Notifying a Waiting Thread of an Event (Cont.)

- Every Java object inherits from the java.lang.Object class:
  - \* - The ability to maintain a queue of threads waiting for an object lock to be released, and
  - \* - To notify one or more waiting threads that the object is freed.
- This provides a great way to notify a thread that an event has occurred, and for threads to wait indefinitely (or for a limited amount of time) until notification is sent.

كل بيوتنا ال notify وال wait

بقدرنا يستريح forever او لوقت محدود

### Notifying a Waiting Thread of an Event (Cont.)

- To have threads wait for an indefinite amount of time, the Object.wait() method is used.
  - \* - An overloaded version of this method also exists, which waits for a limited amount of time (specified in milliseconds).
- Before the wait() method may be invoked, however, the thread must hold a lock on the object's monitor.
  - \* - To gain a lock on an object's monitor, it must be executing a synchronized method or using a synchronized block.
  - \* - When the lock is released, another thread can obtain it—without this, the thread will wait indefinitely.

blocking wait

اللي يجيبنا لخلينا

ال انوب ارم نظرونا synchronize

\* لخلي thread يباش wait لازم ياخذ ال lock على ال object اللي به يستريح كنه

فكل مرة بنستريح notify كايضا بنعمل release ال lock عشان يفوت اللي بجديه ياخذ ال lock وهكذا

بشكل اذق لما يتنفذ ال انوب مجازي synchronized لحتى كل thread لما به يفوت ال state لازم يكون ماخذ ال lock تاغ ال object عشان يفوت حاله و لما يجيبنا بحالة ال monitor بعمل release ال lock ليعمل ل threads تانيه يفوتها بال wait state عن نفس ال object

### Notifying a Waiting Thread of an Event (Cont.)

\* Once the wait() method is executed, the monitor is released and the thread is suspended until a call is made to the Object.notify() or Object.notifyAll() method.

To awaken waiting threads, another thread may call either method.

\* However, the notify() method will only notify a single thread, even if multiple threads are waiting.

\* There is no choice over which thread is awakened, either (this is determined by the JVM implementation, so you cannot rely on, for example, a FIFO queue).

\* It is advised that the notifyAll() method is used if you want to notify a specific thread.

ليس وحدة  
بفعلوا notify  
(مين هويما يعرف)  
لقتعد التنفيذ  
وعالاه

كلمة بفعالنا notify  
كازوم ال thread ال waiting بقتعد بـ we استنوا  
هيك لنكنا

buffer

57

لازم يتنادوا بـ Synchronize

### Notifying a Waiting Thread of an Event (Example)

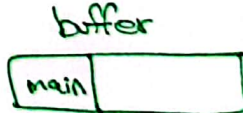
```
public class WaitNotify extends Thread{
    public static void main(String args[]) throws Exception{
        Thread notificationThread = new WaitNotify();
        notificationThread.start();
        // Wait for the notification thread to trigger event
        synchronized (notificationThread){
            notificationThread.wait();
        }
        // Notify user that the wait() method has returned
        System.out.println ("The wait is over");
    }
}
```

بصيف  
ال buffer

ال main ال object الي بقتعد  
تستنى فيه وتناخذ  
ال lock تبقعه

بشكلك عام صحتي  
تعمل على  
object

لما يجيني  
notify



68

### Notifying a Waiting Thread of an Event (Example)

```

public void run(){
    System.out.println ("Hit enter to stop waiting thread");
    try{
        System.in.read();
    }
    catch (java.io.IOException ioe){}
    // Notify any threads waiting on this thread
    synchronized (this){
        this.notifyAll();
    }
}

```

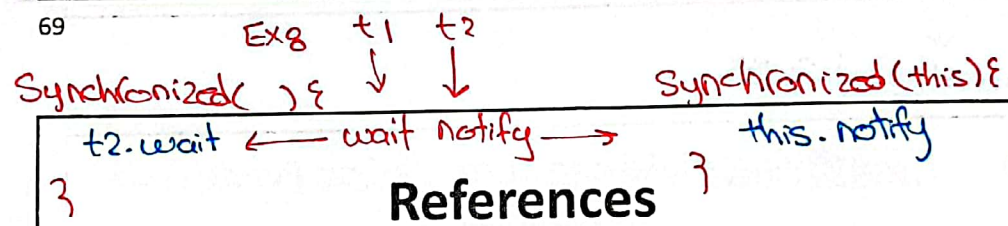
to block until the user enter anything.

بزريلدمن  
buffer JI

نوتيفي  
للكل ال  
اللي  
بيستوي

لما تنتقد  
بطلع اللي  
اللي  
من ال  
وينقد  
the wait is over

نوتيفي  
ال  
thread



### References

Chapter 7 of Java™ Network Programming and Distributed Computing, David Reilly and Michael Reilly.

Chapter 30 of Introduction to Java Programming by Y. Daniel Liang, 10<sup>th</sup> edition.

# Object-Oriented Problem Solving

## Exception Handling

*Based on Chapter 12 of "Introduction to Java Programming" by Y. Daniel Liang.*



Eng. Asma Abdel Karim  
Computer Engineering Department

## Outline

- Introduction (12.1)
- Exception Handling Overview (12.2)
- Exception Types (12.3)
- More on Exception Handling (12.4)
  - Declaring Exceptions (12.4.1)
  - Throwing Exceptions (12.4.2)
  - Catching Exceptions (12.4.3)
  - Getting Information from Exceptions (12.4.4)
  - Example: Declaring, Throwing, and Catching Exceptions (12.4.5)
- The *finally* Clause (12.5)
- When to use Exceptions? (12.6)
- Rethrowing Exceptions (12.7)
- Chained Exceptions (12.8)
- Defining Custom Exception Classes (12.9)



Eng. Asma Abdel Karim  
Computer Engineering Department

### 3 types of errors :

- 1- Syntax errors.
- 2- Runtime errors.
- 3- logical errors

# Introduction

- Runtime errors occur while a program is running if the JVM detects an operation that is impossible to carry out.
  - \* – If you access an array using an index that is out of bounds, you will get a runtime error with an *ArrayIndexOutOfBoundsException*.
  - \* – If you enter a *double* value when your program expects an *integer*, you will get a runtime error with an *InputMismatchException*.
- In Java, runtime errors are thrown as *exceptions*.
- An *exception* is an object that represents an error or a condition that prevents execution from proceeding normally.
- If the *exception* is not handled, the program will terminate abnormally.
  - *Exception handling* enables a program to deal with exceptional situations and continue its normal execution.

خطأ  
الوقت  
Runtime  
error



Eng. Asma Abdel Karim  
Computer Engineering Department

3

## Exception Handling Overview

### Quotient.java

```

1  import java.util.Scanner;
2
3  public class Quotient {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6
7          // Prompt the user to enter two integers
8          System.out.print("Enter two integers: ");
9          int number1 = input.nextInt();
10         int number2 = input.nextInt();
11
12         System.out.println(number1 + " / " + number2 + " is " +
13             (number1 / number2));
14     }
15 }

```

Enter two integers: 5 2   
5 / 2 is 2

Enter two integers: 3 0   
Exception in thread "main" java.lang.ArithmeticException: / by zero  
at Quotient.main(Quotient.java:11)

\* بالجزء القسمة  
منو بتطلبك  
Integer/int فقط لو بقسم  
لو في واحد Float ما بطلع  
error



Eng. Asma Abdel Karim  
Computer Engineering Department

4

# Exception Handling Overview (Cont.)

## QuotientWithIf.java

```
1 import java.util.Scanner;
2
3 public class QuotientWithIf {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         // Prompt the user to enter two integers
8         System.out.print("Enter two integers: ");
9         int number1 = input.nextInt();
10        int number2 = input.nextInt();
11
12        if (number2 != 0)
13            System.out.println(number1 + " / " + number2
14                + " is " + (number1 / number2));
15        else
16            System.out.println("Divisor cannot be zero ");
17    }
18 }
```

عملت ال Code  
بدون exception  
handling

بس هيك ما بزيك لو حد ثاني غيري به بيستخدم ال Code لإنه ما عطيه مجال يختار هو نشو  
ينعمل لو  
مار error

Enter two integers: 5 0

Divisor cannot be zero



Eng. Asma Abdel Karim  
Computer Engineering Department

5

# Exception Handling Overview (Cont.)

## QuotientWithMethod.java

```
1 import java.util.Scanner;
2
3 public class QuotientWithMethod {
4     public static int quotient(int number1, int number2) {
5         if (number2 == 0) {
6             System.out.println("Divisor cannot be zero");
7             System.exit(1);
8         }
9
10        return number1 / number2;
11    }
12
13    public static void main(String[] args) {
14        Scanner input = new Scanner(System.in);
15
16        // Prompt the user to enter two integers
17        System.out.print("Enter two integers: ");
18        int number1 = input.nextInt();
19        int number2 = input.nextInt();
20
21        int result = quotient(number1, number2);
22        System.out.println(number1 + " / " + number2 + " is "
23            + result);
24    }
25 }
```



Eng. Asma Abdel Karim  
Computer Engineering Department

6

# Exception Handling Overview (Cont.) QuotientWithMethod.java (Output)

Enter two integers: 5 3   
5 / 3 is 1

Enter two integers: 5 0   
Divisor cannot be zero

Program is terminated if  
number2 equals 0.

Problem: what if the caller  
should decide whether to  
terminate the program!



Eng. Asma Abdel Karim  
Computer Engineering Department

# Exception Handling Overview QuotientWithException.java

```

1 import java.util.Scanner;
2
3 public class QuotientWithException {
4     public static int quotient(int number1, int number2) {
5         if (number2 == 0)
6             throw new ArithmeticException("Divisor cannot be zero");
7
8         return number1 / number2;
9     }
10
11     public static void main(String[] args) {
12         Scanner input = new Scanner(System.in);
13
14         // Prompt the user to enter two integers
15         System.out.print("Enter two integers: ");
16         int number1 = input.nextInt();
17         int number2 = input.nextInt();
18
19         try {
20             int result = quotient(number1, number2);
21             System.out.println(number1 + " / " + number2 + " is "
22                 + result);
23         } catch (ArithmeticException ex) {
24             System.out.println("Exception: an integer "
25                 + "cannot be divided by zero ");
26         }
27
28         System.out.println("Execution continues ...");
29     }
30 }
31

```

مكتبة داخل ال Class الي فيه  
ال Method الي حيسين فيها Error

→ Part one  
بنويه ك Object من نوع ال error  
الي حيسار .

من هون  
يبقى ال  
Code

→ Part 2 → handling  
the error



Eng. Asma Abdel Karim  
Computer Engineering Department



# Exception Handling Overview

## QuotientWithException.java (Output)

```
Enter two integers: 5 3   
5 / 3 is 1  
Execution continues ...
```

```
Enter two integers: 5 0   
Exception: an integer cannot be divided by zero  
Execution continues ...
```



Eng. Asma Abdel Karim  
Computer Engineering Department

9

# Exception Handling Overview

« بالغالب جنكون بنكتب هذا ال Part »

```
try {  
    Code to run;  
    A statement or a method that may throw an exception;  
    More code to run;  
}  
catch (type ex) {  
    Code to process the exception;  
}
```

→ ينط جواه بين ال Statement الي الهم  
علاقة بال error مش كل اشئ ينط  
→ ممكن يكون أكثر من Catch



Eng. Asma Abdel Karim  
Computer Engineering Department

10

# Exception Handling Overview

## Benefit of Exception Handling

- The key benefit of exception handling is separating the detection of an error (done in a called method) from the handling of an error (done in the calling method).
  - Often the called method does not know what to do in case of error.
  - This is typically the case for the library methods.
    - The library method can detect the error, but only the caller knows what needs to be done when an error occurs.



Eng. Asma Abdel Karim  
Computer Engineering Department

11

# Exception Handling Overview

## InputMismatchExceptionDemo.java

```

1  import java.util.*;
2
3  public class InputMismatchExceptionDemo {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6          boolean continueInput = true;
7
8          do {
9              try {
10                 System.out.print("Enter an integer: ");
11                 int number = input.nextInt();
12                 // Display the result
13                 System.out.println(
14                     "The number entered is " + number);
15
16                 continueInput = false;
17             }
18             catch (InputMismatchException ex) {
19                 System.out.println("Try again. (" +
20                     "Incorrect input: an integer is required)");
21                 input.nextLine(); // Discard input
22             }
23         } while (continueInput);
24     }
25 }
26 }

```

If an  
InputMismatch  
Exception  
occurs

لو ال user دخل اشئ مش  
integer بحير error

\*مشكلة ال nextInt بتقرأ  
integer عالقد يعني لو كبت الرقم

بعدن Enter ما بتستثنيه فبطوع  
error

فرطيت هاي كشان

ما ياخذ الاشئ الي كان معلق لا يرجع ياخذ  
ال input الي حيدخله ال user الجديد.

initially true



Eng. Asma Abdel Karim  
Computer Engineering Department

12

# Exception Handling Overview

## InputMismatchExceptionDemo.java (Output)

```
Enter an integer: 3.5 
Try again. (Incorrect input: an integer is required) ← "catch" ما صلب
Enter an integer: 4 
The number entered is 4
```



## Exception Types

- Exceptions are objects, and objects are defined using classes.
- The root class for all exceptions is *java.lang.Throwable*.
- There are many predefined exception classes in the Java API.
- You can also define your own exception classes.

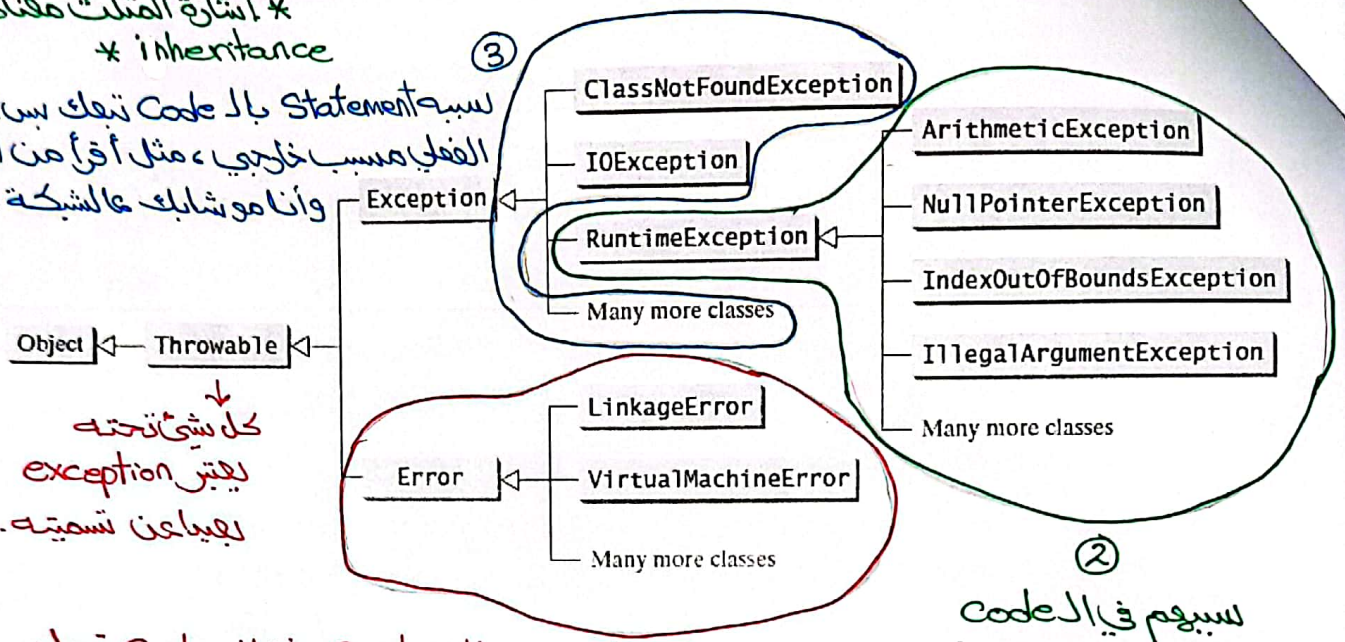
ما محتاج  
هيك بال NP



# Exception Types (Cont.)

\* إشارة الفئتين ههنا  
\* inheritance

لسبب Statement بال Code تنك بس المسبب  
الفعلي مسبب خارجي ، مثل أقر من الشبكة  
وأنا مو شايفك عالشبكة.



كده شئ تحتته  
يعتبر exception  
بهيابن تسميته.

① لسبب ال System من ال Code تنك

② لسببوم في ال code  
قد يكون من المبرمج أو  
من ال user



Eng. Asma Abdel Karim  
Computer Engineering Department

# Exception Types (Cont.)

- The *Throwable* class is the root of all exception classes.
  - \* – All Java exception classes inherit directly or indirectly from *Throwable*.
  - \* – You can create your own exception classes by extending *Exception* or a subclass of *Exception*.
- The exception classes can be classified into three major types:
  - \* – System Errors.
  - \* – Runtime Exceptions.
  - \* – Other exceptions.



Eng. Asma Abdel Karim  
Computer Engineering Department

\* ما بنعملة handle لانها ما بنعرفي بسية  
\* نادر ما يحدث بس قد يحدث

①

## Exception Types: System Errors

- *System errors* are thrown by the JVM and are represented in the *Error* class.
- The *Error* class describes internal system errors, though such errors rarely occur.
  - If one occurs, there is little you can do beyond notifying the user and trying to terminate the program gracefully.

| Class               | Reasons for Exception                                                                                                                  |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| LinkageError        | A class has some dependency on another class, but the latter class has changed incompatibly after the compilation of the former class. |
| VirtualMachineError | The JVM is broken or has run out of the resources it needs in order to continue operating.                                             |



Eng. Asma Abdel Karim  
Computer Engineering Department

17

②

## Exception Types: Runtime Exceptions

- Runtime exceptions are represented in the *RuntimeException* class.
  - Describes programming errors, such as bad casting, accessing an out-of-bounds array, and numeric errors.

| Class                     | Reasons for Exception                                                                                                                               |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| ArithmeticException       | Dividing an integer by zero. <u>Note that floating-point arithmetic does not throw exceptions.</u> (see Appendix E, Special Floating-Point Values). |
| NullPointerException      | Attempt to access an object through a null reference variable.                                                                                      |
| IndexOutOfBoundsException | Index to an array is out of range.                                                                                                                  |
| IllegalArgumentException  | A method is passed an argument that is illegal or inappropriate.                                                                                    |



Eng. Asma Abdel Karim  
Computer Engineering Department

18

③ ← أكثر أنواع استثناءاتك فيه بالـ NP  
**Exception Types: Other Exceptions**

- Other exceptions are represented in the *Exception* class.
  - \* – Describes errors caused by your program and by external circumstances.

| Class                  | Reasons for Exception                                                                                                                                                                                                                                              |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ClassNotFoundException | Attempt to use a class that does not exist. This exception would occur, for example, if you tried to run a nonexistent class using the java command, or if your program were composed of, say, three class files, only two of which could be found.                |
| IOException            | Related to input/output operations, such as invalid input, reading past the end of a file, and opening a nonexistent file. Examples of subclasses of IOException are InterruptedException, EOFException (EOF is short for End of File), and FileNotFoundException. |

**Exception Types: Checked and Unchecked**

← هذا أنا مجبور أخط try و catch دائما؟  
 مو شرط حسب الموقف.  
**Exceptions**

- RuntimeException, Error and their subclasses are known as unchecked exceptions. → يعني بتركلك العربية تفعل handling أو لا.
- \* – In most cases, unchecked exceptions reflect programming logic errors that are unrecoverable.
- \* – To avoid cumbersome overuse of try-catch blocks, Java does not mandate that you write code to catch or declare unchecked exceptions.
- All other exceptions are known as checked exceptions. → لازم تفعلهم handling ما عندك خيار.
- \* – The compiler forces the programmer to check and deal with them in a try-catch block or declare it in the method header.

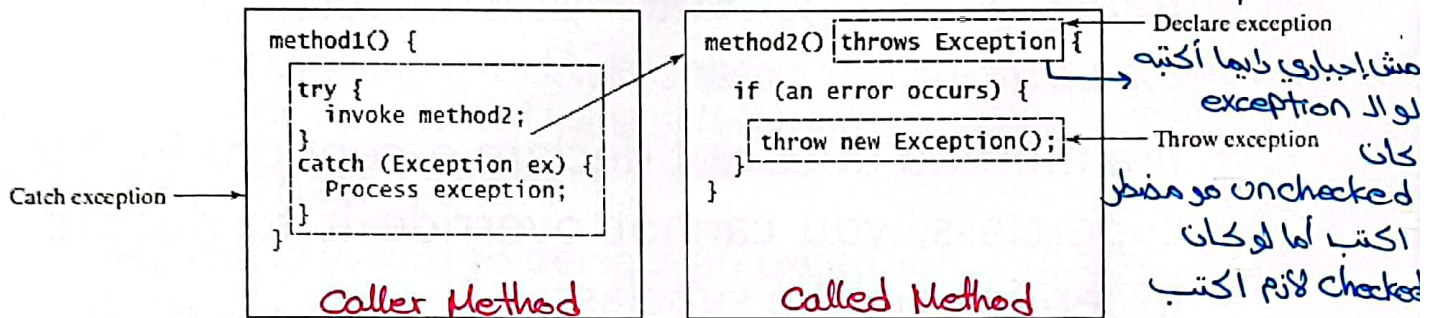
بتقدر تتفادى بدون ما تفعل handling

# More On Exception Handling

- Java exception handling model is based on three operations:

- 1 – Declaring an exception.
- 2 – Throwing an exception. → in *calledMethod*
- 3 – Catching an exception. → in *Caller*

حظيته بالتعريف عشان يكون  
واضح المبرمج هاوال  
exceptions نشو ممكن ترمي  
↑



Eng. Asma Abdel Karim  
Computer Engineering Department

21

# More On Exception Handling

## ① Declaring Exceptions

- Every method must state the types of *checked exceptions* it might throw.

\* – This is known as *declaring exceptions*.

\* – Java does not require that you declare unchecked exceptions explicitly in the method.

- To declare an exception in a method, use the *throws* keyword in the method header.

- Example:

```
public void myMethod() throws IOException
```



Eng. Asma Abdel Karim  
Computer Engineering Department

22

## More On Exception Handling Declaring Exceptions (Cont.)

- If the method might throw multiple exceptions, add a list of the exceptions, separated by commas after throws:

```
public void myMethod() throws Exception1,  
Exception2, ..., ExceptionN
```

- If a method does not declare exceptions in the superclass, you cannot override it to declare exceptions in the subclass. ↓

لازم يكون ال header زي ال Super class



Eng. Asma Abdel Karim  
Computer Engineering Department

23

## More On Exception Handling ② Throwing Exceptions

- A program that detects an error can create an instance of an appropriate exception type and throw it.
  - This is known as *throwing an exception*.

### Example:

Suppose the program detects that a negative argument is passed when it should be nonnegative, the program can create an instance of *IllegalArgumentException* and throw it as follows:

```
IllegalArgumentException ex = new  
IllegalArgumentException ("Wrong Argument");
```

```
throw ex;
```

OR

```
throw new IllegalArgumentException ("Wrong Argument");
```

مثبت  
↓ Throws  
أعلى  
↓ التعرف

\* نخرج جولي Address دائمتا

↓  
" هذا أفضل للاستخدام "



Eng. Asma Abdel Karim  
Computer Engineering Department

24



## More On Exception Handling Throwing Exceptions (Cont.)

- In general, each exception class in the Java API has at least two constructors:
  - A: – A no-arg constructor, and
  - B: – A constructor with a String argument that describes the exception.
    - \*• The argument is called the *exception message*, which can be obtained using `getMessage()`;
- Note that: *access كمان اعل*
  - \* – The keyword to declare an exception is throws.
  - \* – The keyword to throw an exception is throw.



Eng. Asma Abdel Karim  
Computer Engineering Department

25

## More On Exception Handling ③ Catching Exceptions

- When an exception is thrown, it can be caught and handled in a *try-catch* block, as follows:

```
try{
    statements; //statements that may throw exception
}
catch (Exception1 exVar1){ one parameter exactly
    handler for exception1;
}
catch (Exception2 exVar2){
    handler for exception2;
}
...
catch (ExceptionN exVarN){
    handler for exceptionN;
}
```



Eng. Asma Abdel Karim  
Computer Engineering Department

26

## More On Exception Handling Catching Exceptions (Cont.)

- If no exceptions arise during the execution of the *try* block, the *catch* blocks are skipped.
- If one of the statements inside the *try* block throws an exception:
  - \*- Java skips the remaining statements in the *try* block, and
  - \*- Starts the process of finding the code to handle the exception, which is called *catching an exception*.
    - ⊙ The code that handles the exception is called the *exception handler*.



Eng. Asma Abdel Karim  
Computer Engineering Department

27

## More On Exception Handling Catching Exceptions (Cont.)

catch  
block

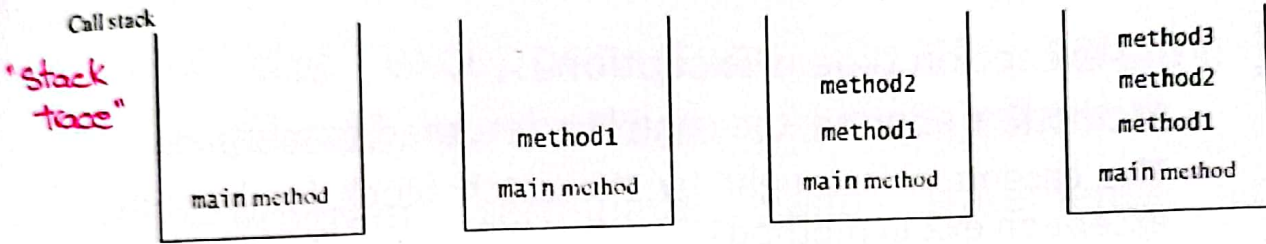
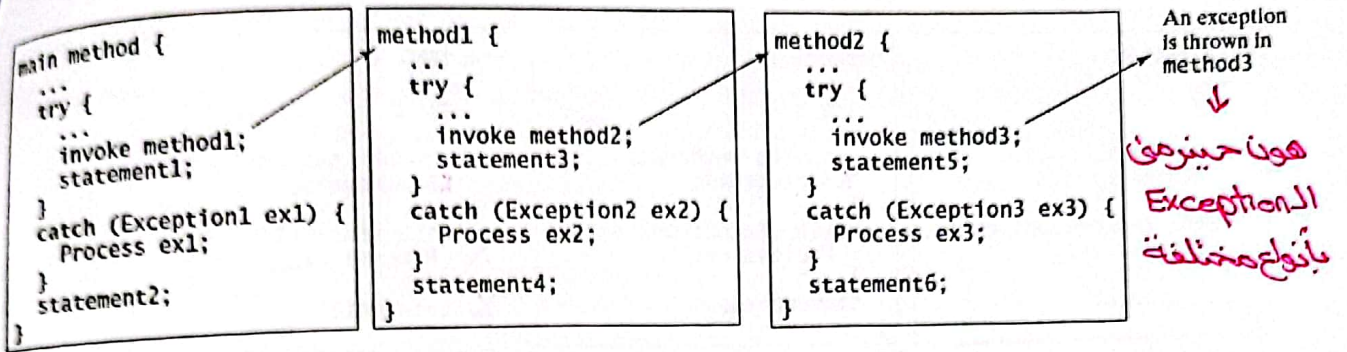
- An exception handler is found by *propagating the exception* backward through a chain of method calls, starting from the current method. → لو كان ما في Catch مطابق للذي انا فيه
- Each catch block is examined in turn, from first to last, to see whether the type of the exception object is an instance of the exception class in the catch block
  - \*- If so, the exception object is assigned to the variable declared, and the code in the catch block is executed.
  - \*- If no handler is found, Java exits this method, passes the exception to the method that invoked the method, and continues the same process to find a handler.
  - \*- If no handler is found in the chain of methods being invoked, the program terminates and prints an error message to the console.



Eng. Asma Abdel Karim  
Computer Engineering Department

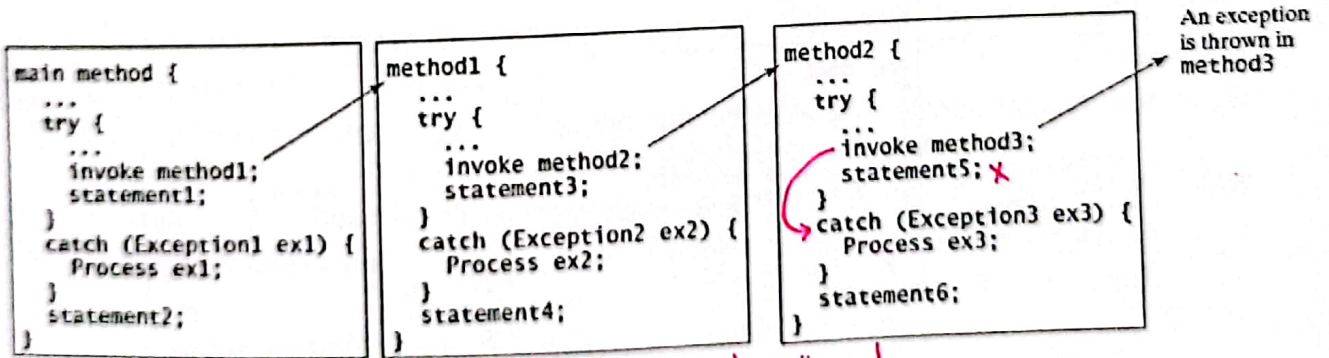
28

# Catching Exceptions: An Example



Eng. Asma Abdel Karim  
Computer Engineering Department

# Catching Exceptions: An Example (Case 1)



ال Exception 3 جينرسي في ال Method 2

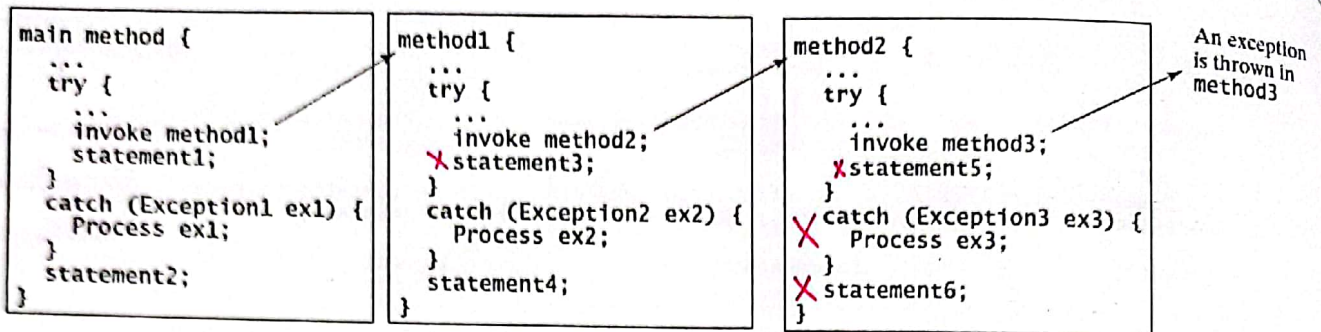
As • If the exception type is Exception3:

- It is caught by the catch block for handling exception ex3 in method2.
- Statement 5 is skipped, and statement6 is executed.



Eng. Asma Abdel Karim  
Computer Engineering Department

## Catching Exceptions: An Example (Case 2)



**B :** If the exception type is Exception2:

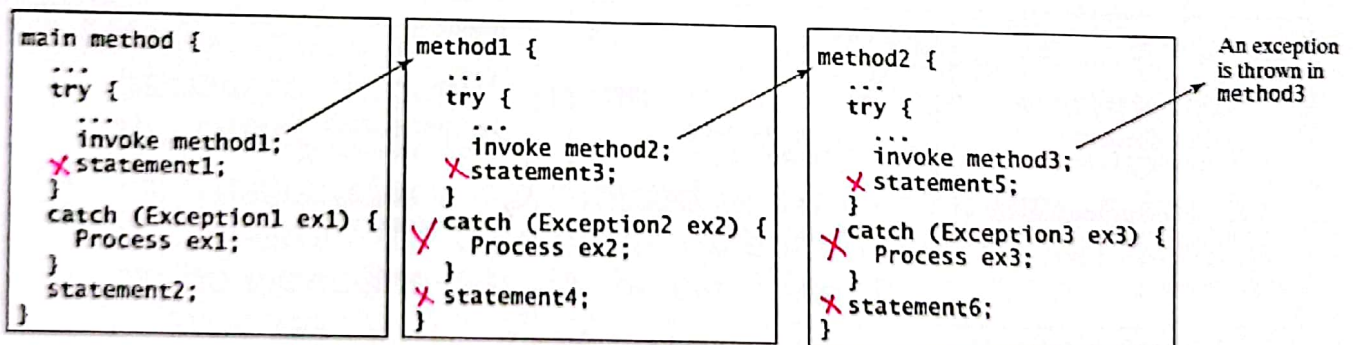
- Method2 is aborted, the control is returned to method1.
- The exception is caught by the catch block for handling exception ex2 in method1.
- Statement3 is skipped, and statement4 is executed.



Eng. Asma Abdel Karim  
Computer Engineering Department

31

## Catching Exceptions: An Example (Case 3)



**C :** If the exception type is Exception1:

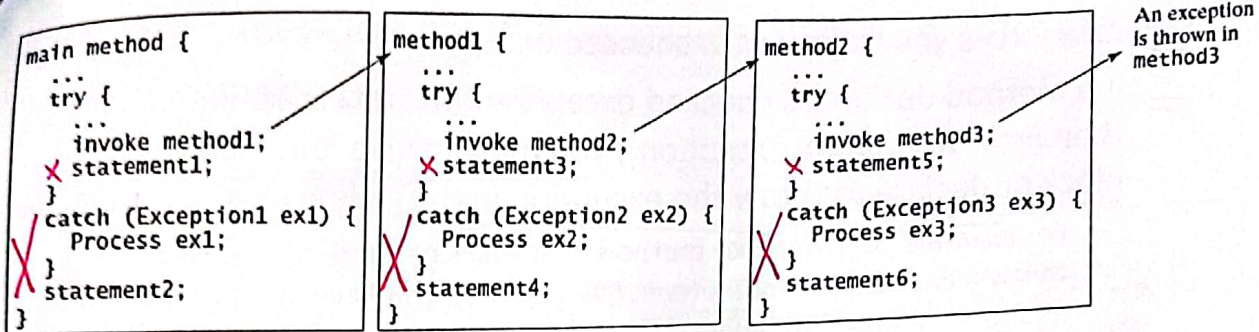
- Method2 and method1 are aborted, the control is returned to the main method.
- The exception is caught by the catch block for handling exception ex1 in the main method.
- Statement1 is skipped, and statement2 is executed.



Eng. Asma Abdel Karim  
Computer Engineering Department

32

# Catching Exceptions: An Example (Case 4)



- If the exception type is not caught in method2, method1, or the main method:
  - \* - Program terminates, and statement1 and statement2 are not executed.



Eng. Asma Abdel Karim  
Computer Engineering Department

33

\* بانما تدرج من الأخص للأعم مثل العكس ، خط ال Sub قبل ال Super.  
\* Variable من النوع العام بقدر يأشر على Object من النوع الخاص.

## More on Catching Exceptions

- Various exception classes can be derived from a common superclass.
  - \* - If a catch block catches exception objects of a superclass, it can catch all the exception objects of the subclasses of the superclass.
- The order in which exceptions are specified in catch blocks is important.
  - \* - A compile error will result if a catch block for a superclass type appears before a catch block for a subclass type.

```

try {
    ...
}
catch (Exception ex) {
    ...
}
catch (RuntimeException ex) {
    ...
}
    
```

(a) Wrong order

```

try {
    ...
}
catch (RuntimeException ex) {
    ...
}
catch (Exception ex) {
    ...
}
    
```

(b) Correct order

لو انوه  
Illegal Argument

بقدر يقطع ال Runtime

لانه هو Subclass من ال Runtime

34



Eng. Asma Abdel Karim  
Computer Engineering Department

أبداً Exception من ال Runtime  
وانما هي تملك هونا.

## More on Catching Exceptions (Cont.)

- Java forces you to deal with checked exceptions.
- If a method declares a checked exception (i.e., an exception other than *Error* or *RuntimeException*), you must invoke it in a try-catch block or declare to throw the exception in the calling method. **A**
  - For example, suppose that method *p1* invokes method *p2*, and *p2* may throw a checked exception (e.g., *IOException*); you have to write the code as shown in (a) or (b) below.

```
void p1() {
    try {
        p2();
    }
    catch (IOException ex) {
        ...
    }
}
```

(a) Catch exception  
"handling"

```
void p1() throws IOException {
    p2();
}
```

(b) Throw exception



Eng. Asma Abdel Karim  
Computer Engineering Department

لوما بيك تفعل handle على الأقل  
اكتب بالتعريف انه في exception

35

## More on Catching Exceptions (Cont.)

- You can use the new JDK 7 multi-catch feature to simplify coding for the exceptions with the same handling code.
- The syntax is:

```
catch (Exception1 | Exception2 | ... | Exceptionk ex) {
    // Same code for handling these exceptions
}
```

\* مفيدة لما جلد كل ال exception تكون نفس ال Code



Eng. Asma Abdel Karim  
Computer Engineering Department

36

# Getting Information from Exceptions

- An exception object contains valuable information about the exception.

| java.lang.Throwable                   |                                                                                                                                                   |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| +getMessage(): String                 | Returns the message that describes this exception object.                                                                                         |
| +toString(): String                   | Returns the concatenation of three strings: (1) the full name of the exception class; (2) ":" (a colon and a space); (3) the getMessage() method. |
| +printStackTrace(): void              | Prints the Throwable object and its call stack trace information on the console.                                                                  |
| +getStackTrace(): StackTraceElement[] | Returns an array of stack trace elements representing the stack trace pertaining to this exception object.                                        |

Stacktrace ← شوال Method التي ال Exception مرق فيها لحد ما انلق وانحل .

- Name of Method
- Class
- File
- Line

37



Eng. Asma Abdel Karim  
Computer Engineering Department

## TestException.java

```

1 public class TestException {
2     public static void main(String[] args) {
3         try {
4             System.out.println(sum(new int[] {1, 2, 3, 4, 5}));
5         }
6         catch (Exception ex) {
7             ex.printStackTrace();
8             System.out.println("\n" + ex.getMessage());
9             System.out.println("\n" + ex.toString());
10
11             System.out.println("\nTrace Info Obtained from getStackTrace");
12             StackTraceElement[] traceElements = ex.getStackTrace();
13             for (int i = 0; i < traceElements.length; i++) {
14                 System.out.print("method " + traceElements[i].getMethodName());
15                 System.out.print("(" + traceElements[i].getClassName() + "):");
16                 System.out.println(traceElements[i].getLineNumber() + "");
17             }
18         }
19     }
20
21     private static int sum(int[] list) {
22         int result = 0;
23         for (int i = 0; i <= list.length; i++)
24             result += list[i];
25         return result;
26     }
27 }
    
```

رجعت Exception!

عم نطلع بتر  
حدود ال Array  
هو حد تظهر  
المشكلات



Eng. Asma Abdel Karim  
Computer Engineering Department

38

# TestException.java (Output)

← *ال index الذي يسبب exception array of bound JUM خطته*

```

C:\book>java TestException
java.lang.ArrayIndexOutOfBoundsException: 5
    at TestException.sum(TestException.java:24)
    at TestException.main(TestException.java:4)
5
java.lang.ArrayIndexOutOfBoundsException: 5
Trace Info Obtained from getStackTrace
method sum(TestException:24)
method main(TestException:4)
C:\book>_
    
```

← *toString*  
← *اللي صار فيها ال exception*  
← *اللي كانت*  
← *Sum ال*  
← *من*  
← *line 11 → 16*  
← *في ال Code*  
← *أنا زبطتها وعلقتها*

← *printStackTrace()*  
← *getMessage()*  
← *toString()*  
← *Using getStackTrace()*



Eng. Asma Abdel Karim  
Computer Engineering Department

39

## Example: Declaring, Throwing, and Catching Exceptions (CircleWithException.java)

```

public class CircleWithException{
    private double radius;
    private static int numberOfObjects=0;

    public CircleWithException(double newRadius){
        setRadius(newRadius);
        numberOfObjects++;
    }

    public void setRadius(double newRadius) throws IllegalArgumentException{
        if (newRadius>=0) radius = newRadius;
        else throw new IllegalArgumentException("Radius cannot be negative!");
    }

    public static int getNumberOfObjects(){
        return numberOfObjects;
    }
}
    
```



Eng. Asma Abdel Karim  
Computer Engineering Department

40



## Example: Declaring, Throwing, and Catching Exceptions (TestCircleWithException.java)

```
public class TestCircleWithException{
    public static void main (String [] args){
        try{
            CircleWithException C1 = new CircleWithException(5);
            CircleWithException C2 = new CircleWithException(-5);
            CircleWithException C3 = new CircleWithException(0);
        }
        catch (IllegalArgumentException ex){
            System.out.println(ex);
        }
        System.out.println("Number of circle objects created: "+
            CircleWithException.getNumberOfObjects());
    }
}
```

**Output:**  
java.lang.IllegalArgumentException: Radius cannot be negative!  
Number of circle objects created: 1



Eng. Asma Abdel Karim  
Computer Engineering Department

41

## ← بنظروا بعد ال Catch بنشغل دائما شو ما صار The *finally* Clause

- The *finally* clause is executed under all circumstances, regardless of whether an exception occurs in the try block or is caught.
- The syntax for the *finally* clause is as follows:

```
try {
    Statements
}
catch (TheException ex){
    handling ex;
}
finally{
    finalStatements;
}
```



Eng. Asma Abdel Karim  
Computer Engineering Department

42

## The *finally* Clause (Cont.)

- If no exception arises in the *try* block:
  - \* – The *finally* clause is executed, and
  - \* – The next statement after the *try* statement is executed.
- If a statement causes an exception in the *try* block that is caught in the *catch* block:
  - \* – The rest of the statements in the *try* block are skipped,
  - \* – The *catch* block is executed,
  - \* – The *finally* clause is executed, and
  - \* – The next statement after the *try* statement is executed.
- If a statement causes an exception that is not caught in any *catch* block:
  - \* – The other statements in the *try* block are skipped,
  - \* – The *finally* clause is executed, and
  - \* – The exception is passed to the caller of this method.
- Note: the *finally* block executes even if there is a *return* statement prior to reaching the *finally* block.



Eng. Asma Abdel Karim  
Computer Engineering Department

43

بجملتها حد ثاني بيده  
يستخدم ال code فيري

## When to Use Exceptions?

- The *try* block contains the code that is executed in normal circumstances.
- The *catch* block contains the code that is executed in exceptional circumstances.
- Exception handling separates error-handling code from normal programming tasks, thus making programs easier to read and to modify.
- Be aware, however, that exception handling usually requires more time and resources.
  - \* – Requires instantiating a new exception object,
  - \* – Rolling back the call stack, and
  - \* – Propagating the exception through the chain of methods invoked to search for the handler.



Eng. Asma Abdel Karim  
Computer Engineering Department

44

## When to Use Exceptions? (Cont.)

- An exception occurs in a method:
  - If you want the exception to be processed by the method's caller, you should create an exception object and throw it.
  - If you can handle the exception in the method where it occurs, there is no need to throw or use exception objects.
    - Simple errors that may occur in individual methods are best handled without throwing exceptions.
    - This can be done by using *if* statements to check for errors.