

**The University of Jordan**  
**Computer Engineering Department**  
**Object Oriented Problem Solving (CPE 342)**  
**Final Exam – Spring 2014**  
**(100 Minutes)**

الرقم الجامعي:

الاسم الثلاثي:

**Question 1:** In the following multiple choice questions, identify the choice that represents the correct answer. (10 points)

1. Which of the following packages is automatically loaded, so you don't need to import it?
  - a. java.util
  - b. javax.swing
  - c. java.awt
  - d. java.lang
  - e. java.math
  
2. Which limitation(s) of arrays do ArrayLists overcome?
  - a. Arrays cannot be searched; ArrayLists can.
  - b. Arrays cannot increase in size; ArrayLists can.
  - c. Arrays cannot be passed as parameters to methods; ArrayLists can.
  - d. There is a method that returns the length of an ArrayList; whereas there is no way to find the length of an array.
  - e. a & b
  
3. To what object-oriented programming concept does the Java keyword *private* apply?
  - a. Polymorphism
  - b. Inheritance
  - c. Encapsulation
  - d. Overriding
  - e. a & b

4. Consider the following class definition:

```
1 public class TimeRecord {
2     private int hours;
3     private int minutes;
4
5     public TimeRecord(){
6         hours=12;
7         minutes=30;
8     }
9
10    //Other methods
11
12    //Adds h hours and m minutes to this TimeRecord
13    public void advance(int h, int m){
14        hours += h;
15        minutes += m;
16        //Missing code
17    }
18 }
```

The instance variable `minutes` must always be at least 0 and less than 60, even when the time is changed. Which of the following code sections can be used to replace the missing code (line 16) in the `advance()` method so that this condition remains true and the time is correctly advanced?


- a. `hours += minutes/60;`  
`minutes %= 60;`
- b. `minutes %= 60;`
- c. `hours += minutes%60;`  
`minutes /= 60;`
- d. `minutes /= 60;`
- e. `minutes += hours%60;`

5. Which of the following is true regarding exceptions in Java?

- a. The compiler forces the programmer to deal with unchecked exceptions.
- b. System errors are example of checked exceptions.
- c. An exception handler cannot rethrow a caught exception to the calling method.
- d. All Java exception classes inherit directly or indirectly from *Throwable* which is a subclass from the *Object* class.
- e. A method in a subclass can be overridden to declare an exception even if it was not declared in its superclass.

6. Consider the following class definition:

```
public class G {
    public static G doStuff1(){
        G newG = new G();
        return newG;
    }
    public static void doStuff2(){
        G newG = new G();
    }
    public static void main(String [] args){
        G g1;
        String s = "Mid";
        G g2 = new G();
        G g3 = g2;
        g1 = doStuff1();
        doStuff2();
        s = "Final";

        
        //Other statements
    }
}
```

How many objects will be eligible for garbage collection at point A?

- a. 0
  - b. 1
  - c. 2
  - d. 3
  - e. 4
7. For the following code segment, which of the statements below evaluates to true?

```
String s1 = new String("Calm down!");
String s2 = "Calm down!";
String s3 = "Calm"+" down!";
char [] charArray = {'C','a','l','m',' ','d','o','w','n','!'};
String s4 = new String(charArray);
```

- a. s1==s2
- b. s2==s3
- c. s2==s4
- d. s1.equals(s2)
- e. b & d

8. Consider the following class definition:

1	public class Class1 {
2	private int x;
3	
4	public void displayX(){
5	System.out.println(x);
6	}
7	
8	class Class2{
9	public void incrementX(){
10	x++;
11	}
12	}
13	}

Which of the following statements is true?

- a. The two classes will be compiled into one class file named Class1\$Class2.class.
  - b. The public modifier can be added to the definition of class2 at line 8.
  - c. A compilation error will be caused at line 10 since Class2 cannot reference private data fields of Class1.
  - d. a & b
  - e. a & c
9. Consider the following classes, which methods show an example of methods overloading and which methods show an example of methods overriding?

- a. There are no overloaded methods, example of methods overriding: method2 (lines 5 & 13).
- b. There are no overloaded methods, example of methods overriding: method1 (lines 2 & 10) and method2 (lines 5 & 13).
- c. Example of methods overloading: method1 (lines 2 & 10)  
Example of methods overriding: method 2 (lines 5 & 13)
- d. Example of methods overloading: method1 (lines 2 & 10), there are no overridden methods.
- e. Example of methods overloading: method1 (lines 2 & 10) and method2 (lines 5 & 13), there are no overridden methods.

1	public class superClass {
2	public void method1(int x, int y){
3	System.out.println(x+" "+y);
4	}
5	private int method2(){
6	return 1;
7	}
8	}
9	class subClass extends superClass{
10	public void method1(){
11	System.out.println("Hi!!");
12	}
13	public int method2(){
14	return 2;
15	}
16	}

10. Which of the following statements is not true about protected data members?

- a. They can be accessed from the same class.
- b. They can be accessed from subclasses in the same package.
- c. They can be accessed from subclasses in different packages.
- d. They can be accessed from all other classes in the same package.
- e. None of the above.

**Question 2: Show the exact outputs of the following programs. (8 points)**

a. **Program 1 (3 points)**

```
public class Point2D {
    private double x=0;
    private double y=0;
    public Point2D(){
    }
    public Point2D(double x, double y){
        this.x=x;
        this.y=y;
    }
    public double getX(){return x;}
    public double getY(){return y;}

    public static void main(String[] args) {
        Object [] points = new Object [5];
        points[0] = new Point2D();
        points[1] = new Point2D(0,0);
        points[2] = new Point3D((Point2D)points[0],0);
        points[3] = new Point3D();

        System.out.println(points[0].equals(points[1]));
        System.out.println(points[2].equals(points[0]));
        System.out.println(points[2].equals(points[3]));
    }
}

class Point3D extends Point2D{
    private double z=0;

    public Point3D(){
    }
    public Point3D(Point2D p, double z){
        super(p.getX(),p.getY());
        this.z=z;
    }
    public boolean equals (Object o){
        if (o instanceof Point3D)
            return (getX()==((Point3D)o).getX() && getY()==((Point3D)o).getY() && z ==((Point3D)o).z);
        else if (o instanceof Point2D)
            return (getX()==((Point2D)o).getX() && getY()==((Point2D)o).getY());
        else return false;
    }
}
}
```

**Output:**

**b. Program 2 (5 points)**

```
import java.util.ArrayList;

class Student{
    private String name;
    private double GPA;
    public static int numOfStudents=0;

    public Student (String name){this.name=name;}

    public String toString (){
        return name + " " + GPA;
    }

    public double getGPA(){return GPA;}

    public void setGPA(double GPA){
        if (GPA>4 || GPA<0) throw new IllegalArgumentException("GPA should be between 0-4.");
        else this.GPA=GPA;
    }
}

public class MainClass {

    public static void method2(ArrayList <Student> list, Student [] array, int index){
        list.add(array[index]);
    }

    public static double method1(ArrayList <Student> list, Student [] array){
        double sum =0;
        try {
            for (int i=0; i<=array.length; i++){
                method2(list, array, i);
                sum += array[i].getGPA();
            }
            System.out.println("Done copying array to array list.");
        }
        catch (IndexOutOfBoundsException ex){
            System.out.println("Error occured while accessing the array.");
        }
        catch (RuntimeException ex){
            System.out.println(ex);
        }
        finally{
            double averageGPA = sum/Student.numOfStudents;
            return averageGPA;
        }
    }
}
```

```
public static void main(String[] args) {
    Student [] students = new Student [3];
    students [0] = new Student ("Ahmed");
    students [1] = new Student ("Sarah");
    students [2] = new Student ("Noor");

    try{
        students[0].setGPA(3.8);
        students[1].setGPA(4.5);
        students[2].setGPA(3.0);
    }
    catch (IllegalArgumentException ex){
        System.out.println(ex);
    }

    ArrayList <Student> studentList= new ArrayList();
    try {
        System.out.println("Average students GPAs: "+method1(studentList, students));
    }
    catch (RuntimeException ex){
        System.out.println("A RuntimeException occured.");
    }
    System.out.println(studentList);
}
}
```

**Output:**

**Question 3: The following code contains 6 syntax errors. In the table below, list the line where the error occurs and the cause of the error. (6 points)**

1.	public class MainClass {
2.	public static void main(String [] args){
3.	Q1 [] q = new Q1 [2];
4.	q[0] = new Q1(1, 1.5);
5.	q[1] = new Q2(4);
6.	
7.	System.out.println(q[1].difference());
8.	System.out.println(q[1].zSquared());
9.	}
10.	}
11.	abstract class Q1 {
12.	private int x;
13.	private double y;
14.	
15.	public Q1 (int x, double y){
16.	this.x=x;
17.	this.y=y;
18.	}
19.	private String toString (){
20.	return x + " " + y;
21.	}
22.	public abstract int difference();
23.	}
24.	class Q2 extends Q1{
25.	public int z;
26.	public Q2(int z){
27.	this.z=z;
28.	}
29.	public int add(){
30.	return x + z;
31.	}
32.	public double zSquared (){
33.	return z*z;
34.	}
35.	}



	Line number	Cause of Error
	1.	
	2.	
	3.	
	4.	
	5.	
	6.	

**Question 4:** You are required to design an application to model a hospital using object-oriented design principles that you learned in this course. (6 points)

The main features of the hospital are as follows:

1. The hospital has a name, address, patients, and departments.
2. Each department has a name and a staff (team members).
3. The application should allow the users to add and remove patients from the hospital system.
4. The application should allow the users to add and remove team members from a specific department in the hospital.
5. A team member could be a doctor or a nurse.
6. Each team member has a name, ID, gender (male or female), and the date he/she joined the hospital.
7. All team members have a maximum working time of 12 hours.
8. Each doctor has a specialty (تخصص) and there are three types of doctors: interns (متدرب), senior doctors, and surgeons (جراح).
9. Each intern has a senior doctor as his/her supervisor.
10. Each senior doctor and surgeon has a group of patients that he/she treats.
11. The application should allow all doctors to check the report of a specific patient.
12. The application should have a method that models that a doctor treats a patient. However interns, senior doctors and surgeons treat patients in a different way, hence they should have different implementations for this method
13. Each patient has a name, birth date, gender (male or female), date he/she was accepted to the hospital, a report that includes the diagnosis made by his doctor, the doctor treating him/her, and number of days he/she will stay in the hospital.
14. All data fields should be encapsulated.

