الإســـــم :	
الرقم الجامعي :	

Before you start the exam, please read the following instructions carefully:

1. Fill the required information above. Your exam will not be graded if any field is empty.
2. This exam booklet has **11 numbered** pages and **4 problems**. Check that your exam includes 10 different pages. Show all of your work on these pages only.
3. This is a closed book and notes exam and no calculators or cell phones are allowed.
4. This exam is for **90 minutes**. The points for each problem are listed to best assist you in managing your time.
5. Make sure to be organized and clear in presenting your solution.

Question	Points	Score
1	**10**	
2	**6**	
3	**4**	
4	**5**	
Lab	**15**	
Total	**40**	

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

1. **Which of the following object oriented concepts means that "a method can be defined without an implementation"?**

    a. Encapsulation.

    b. Inheritance.

    c. Polymorphism.

    d. Overriding.

    e. Abstraction.

2. **Which of the following describes how visibility increases for the accessibility modifiers in Java?**

    a. Private, protected, default, public

    Visibility increases →

    b. Private, default, protected, public

    Visibility increases →

    c. Public, protected, default, private

    Visibility increases →

    d. Public, default, protected, private

    Visibility increases →

    e. Default, private, protected, public

3. **Which of the following <u>does not</u> prevent a method from being overridden?**

    a. Using the *final* keyword in the method definition.

    b. Using the *abstract* keyword in the method definition.

    c. Using the *private* keyword in the method definition.

    d. Using the *static* keyword in the method definition.

    e. c & d.

4. What is the output of the following code segment?

```
String s1 = new String("Fall 2017");
String s2 = "Fall 2017";
String s3 = "Fall "+2017;
String s4 = s3.toLowerCase();

System.out.print((s1==s2)+ ", ");
System.out.print((s2==s3) + ", ");
System.out.print(s1.equals(s2)+ ", ");
System.out.print((s3==s4) + ", ");
System.out.println(s2.equalsIgnoreCase(s4));
```

a. true, true, true, false, true

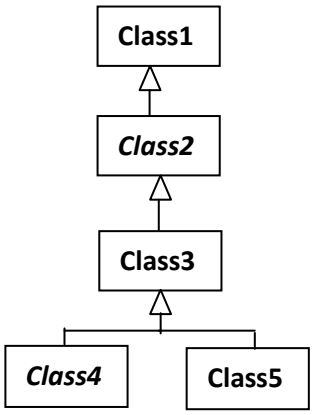
b. false, true, true, false, false

c. false, false, true, false, true

d. false, true, true, false, true

e. true, true, true, true, true

5. According to the following inheritance hierarchy, and assuming that Class2 and Class4 are abstract, which of the following is <u>true</u>?



a. Class1 should have been declared abstract.

b. The following statement causes a compilation error:

   Class2 [] c = new Class2 [4];

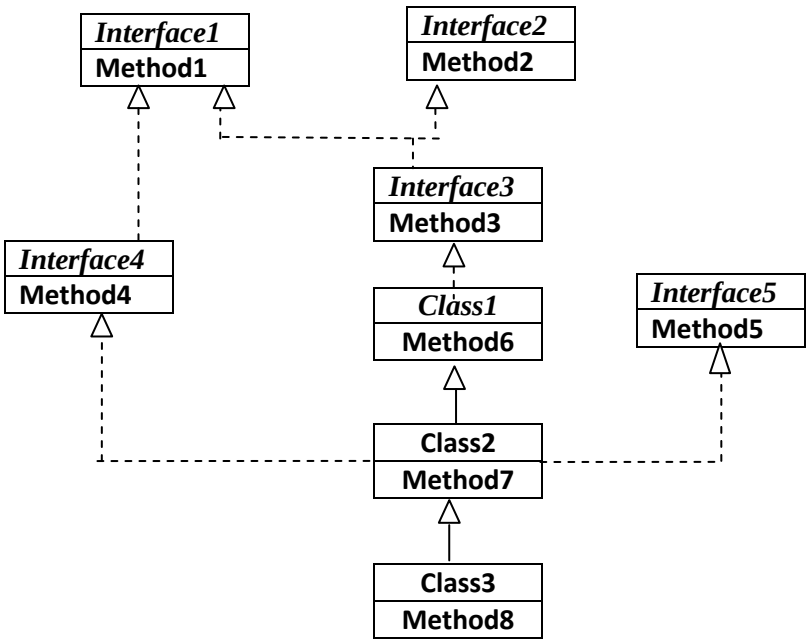
c. Class4 must have at least one abstract method.

d. The following statement causes a compilation error:

   Class4 c = new Class4();

e. None of the above.

Use the following inheritance hierarchy in order to answer questions 6, 7, 8, and 9.



6. **Assume that *Class1* is abstract and implements *Method1*, *Method2* and *Method6* only. Which methods *Class3* must implement, assuming that *Class2* and *Class3* are concrete?**

   a. Method8 only.

   b. Method7 and Method8.

   c. Method3 and Method8

   d. Method3, Method7, and Method8.

   e. Method3, Method4, Method5, Method7, and Method8.

7. **Given the following statement:**

   Class1 c = new Class2();

   Which of the following statements evaluate to false?

   a. c instanceof Interface3

   b. c instanceof Interface1

   c. c instanceof Class2

   d. c instanceof Class3

   e. b & d

8. **When writing a code that implements this hierarchy in Java, what will be the definitions of *Inetrface3* and *Class2*?**

   a. interface Interface3 implements Interface1, Interface2

      class Class2 extends Class1 implements Interface4, Interface5

   b. interface Interface3 extends Interface1, Interface2

      class Class2 extends Class1 implements Interface4, Interface5

   c. interface Interface3 implements Interface1 extends Interface2

      class Class2 extends Class1 implements Interface4, Interface5

   d. interface Interface3 extends Interface1, Interface2

      class Class2 extends Class1 implements Interface1, Interface2, Interface3, Interface4, Interface5

   e. It cannot be implemented in Java since Java does not allow multiple inheritance.

9. **Given the following statement:**

   Class1 c = new Class3();

   Which of the following <u>does not</u> cause a syntax error?

   a. c.Method7()

   b. c.Method8()

   c. c.Method4()

   d. c.Method1()

   e. None of the above statements causes a syntax error

10. **Which of the following is true regarding exception handling in Java?**

    a. The compiler forces the programmer to handle all exceptions declared in the method header.

    b. All exception classes are subclasses from the *Throwable* class but not the *Object* class.

    c. Only checked exceptions should be declared in the method header.

    d. The *finally* clause does not execute if an exception is thrown and not caught by one of the catch blocks.

    e. a & c

<u>Question 2 (6 points):</u> Study the following program. Then, answer the questions that follow.

```
interface Comparable{
   String compareTo(Object o);
}
interface Printable{
   String howToPrint();
}
interface Readable extends Printable{
   String howToRead();
}
public class BookShop implements Comparable{
   Readable[] readables;

   public BookShop(Readable []r){
      readables=new Readable[r.length];
      for(int i=0; i<r.length; i++)
         readables[i]=r[i];
   }
   @Override
   public String compareTo (Object O){
      if (O instanceof BookShop){
         return ""+(readables.length-((BookShop)O).readables.length);
      }
      else return null;
   }
}
abstract class Book implements Readable{
   String name;
   boolean leftToRight;

   public Book(String name, boolean leftToRight){
      this.name=name;
      this.leftToRight=leftToRight;
   }
   @Override
   public String howToRead(){
      if(leftToRight) return "This book is read from left to right";
      else return "This book is read from right to left";
   }
}
class Novel extends Book{
   String authorName="Unknown";

   public Novel(String name, boolean leftToRight, String authorName){
      super(name, leftToRight);
      this.authorName=authorName;
   }
   public Novel(Book b){
      super(b.name, b.leftToRight);
      if (b instanceof Novel) authorName=((Novel)b).authorName;
   }
```

```
    public String howToRead(boolean both){
        if (both) return "This novel has both an english and arabic version!";
        else return "This novel is available in English only";
    }
    @Override
    public String howToPrint(){
        return "Novels should be printed in colors.";
    }
    @Override
    public String toString(){
        return name+"--"+authorName;
    }
    public void displayInfo(){System.out.println(this);}
}
class EnglishNovel extends Novel{
    public EnglishNovel(){
        super("unknown",true,"unknown");
    }
    @Override
    public String toString(){
        return "English Novel: "+super.toString();
    }

}
class Magazine implements Readable{
    String name;

    public Magazine(String name){this.name=name;}
    @Override
    public String howToPrint(){
        return "Print in black and white";
    }
    @Override
    public String howToRead(){
        return "Read it the way you like!";
    }
    @Override
    public String toString(){
        return howToPrint()+"--"+howToRead();
    }
}
```

a. How many .class files will be generated when this file is compiled? (1 point)

b. Why is the Book class declared abstract? (1 point)

c. Given the following objects declaration and creation, fill in the table below with the output of each statement. If the statement causes an error, identify the type and cause of the error. (4 points)

```
Novel n1=new Novel("The Kite Runner", true, "Khaled Husseini");
Novel n2=new EnglishNovel();
Book [] b = {new Novel(n1), new Novel(n2), new EnglishNovel()};
Readable [] r= {b[0], b[2], new Magazine("Alrai"), new Magazine("Aldustoor")};
BookShop bs1 = new BookShop(r);
BookShop bs2 = new BookShop((Readable[])b);
```

Statement	Output
n2.displayInfo();	
System.out.println(bs1.compareTo(bs2));	
System.out.println(n1.howToRead(true));	
bs1.readables[0].displayInfo();	

Question 3 (4 points): **What will the following program print to the console when executed?**

```
public class MainClass {
   public static void main(String[] args) {
      method1();
      System.out.println("Terminating main.");
   }

   public static void method1 (){
      try{
          method2(5);
          System.out.println("Done with method2 invocation.");
      }
      catch(ArithmeticException AE){
         System.out.println(AE);
      }
   }

    public static void method2 (int z){
      try{
         method3(z);
         for (int x=0; x<z; x++)
            System.out.println(x);
         System.out.println("Done printing numbers.");
      }
      catch(ArithmeticException AE){
         System.out.println(AE);
      }
      catch(RuntimeException RE){
         System.out.println("RuntimeException occured");
      }
      System.out.println("Terminating method2.");
   }

     public static void method3(int z){
      try{
         for (int x=0; x<z; x++){
            if (x == 0)
               throw new ArithmeticException ("Divisor cannot be zero!");
            else
              System.out.println(z/x);
         }
         System.out.println("Done printing quotients.");
      }
      catch(IndexOutOfBoundsException ioobe){
         System.out.println(ioobe);
      }
      finally{
         System.out.println("Finally done with exception handling.");
      }
      System.out.println("Terminating method3.");
   }
}
```

Question 4 (5 points): **You are required to design an application to model a cooking contest (مسابقة طبخ) which has a group of chefs (chef:طباخ) participating in the contest, a date that represents the contest date, and a prize which is a certain amount of money.**

The application should be designed as follows:

1. **Each chef has an ID and a group of recipes (وصفات) with which he's participating in the contest. There are two types of chefs: junior chefs and senior chefs. Each senior chef has a number of years that represent his cooking experience. All senior chefs can participate with a maximum of 3 recipes. Whereas junior chefs can participate with one recipe only. Each junior chef must be assigned a senior chef as a supervisor.**
2. **Each recipe has a name and a description of its ingredients (مكونات) and instructions (تعليمات الطبخ).**
3. **Both recipes and chefs can be rated (تقييم) (i.e. they are both ratable). Hence, your design should provide methods to rate chefs and recipes. However, note that senior chefs are rated in a different way from junior chefs. Moreover, it is not allowed to have any chef in the contest that is not specified to be junior or senior.**
4. **Your design should override the equals method of the object class for both chefs and recipes.**
5. **Your design should allow the user to print the rate of all chefs and recipes in the contest.**

Draw the UML class diagram of your design. There's no code writing required in this question.