Information and Computer Science Department

King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Information and Computer Science Department
First Semester (081)
ICS 201 - Introduction to Computing II

Major Exam 1
Monday, 1st December, 2008
Time: 100 minutes

Name:

ID#:

Please circle your section number below:

Section	03	04	05	06
Instructor	Helmy	Ghouti	Sukairi	Yahyaoui
Day and Time	SMW 9 - 9:50	SMW 10 - 10:50	SMW 8 - 8:50	SMW 1 - 2

Question #	Maximum Mark	Obtained Mark
1	20	
2	20	
3	15	
4	10	
5	15	
6	20	
Total	**100**	

Question 1 [Interfaces/Abstract Classes] (4*5 = 20 Marks)

You are designing a graphical Java software application that manipulates different kinds of shapes. The manipulated shapes can be a squares, circles or cubes. The application needs to compute the area and perimeter of a drawn shape.

1. Propose suitable classes and class hierarchies with proper instance variables and method definitions that will fulfill the requirements above.

2. Provide an implementation of the method **computeArea()** for the square, circle and cube classes knowing that the area of a:

   - square: x^2 where x is the length of one of its sides.
   - circle: $\Pi * radius^2$.
   - cube: $6 * a$ where a is the area of one of its associated squares.

3. Provide an implementation of the method **computePerimeter()** for the square, circle and cube classes knowing that the perimeter of a:

   - square: *4*x* where *x* is the length of one of its sides.
   - circle: *2*Π*Radius*.
   - cube: *3*p* where *p* is the perimeter of one of its associated squares.

4. Will the following code fragment compile/run correctly? Explain.

```
Square sq = new Square();
Cube cb = (Cube)sq;
```

Question 2 [Inheritance and Polymorphism] (4*5 = 20 Marks)

Consider the following **Sale** class:

```
public class Sale
{
   private String name; //nonempty string
   private double price; //nonnegative

public Sale() {};
public Sale(String theName, double thePrice)   {
      setName(theName);
      setPrice(thePrice);
   }

   public static void announcement( )    {
      System.out.println("This is the Sale class.");
   }

   public double getPrice( )    {
      return price;
   }

   public void setPrice(double newPrice)    {
      if (newPrice >= 0)
         price = newPrice;
      else
      {
         System.out.println("Error: Negative price.");
         System.exit(0);
      }
   }

   public String getName( )    {
      return name;
   }

   public void setName(String newName)    {
      if (newName != null && newName != "")
         name = newName;
      else
      {
         System.out.println("Error: Improper name value.");
         System.exit(0);
      }
   }

   public String toString( )    {
      return (name + " price = SAR " + price);
   }

 public double bill( )    {
      return price;
   }
}
```

We want to provide an implementation of a class **DiscountSale** which represents a sale having a discount that is represented as a percentage.

1. Provide an implementation of a constructor for the **DiscountSale** class that initializes the fields of that class.

2. Using the overriding concept, provide a method **bill()** that returns a DiscountSale bill, **toString()** that displays: the name, the discounted price and discount of a **DiscountSale** Object, and a method **announcement()** that returns the string "This is a DiscountSale class".

3. Assume that the default constructor of **DiscountSale** is defined. What is the output of the following code?

```
 Sale sl = new DiscountSale("map",5,0);
 DiscountSale ds = new DiscountSale();
 if (sl instanceof DiscountSale) {
     ds = (DiscountSale)sl;
     System.out.println("ds was changed to " + sl);
}
```

4. Suppose you add the following method to the class **Sale**:

```
public void showAdvertisement() {
    announcement();
    System.out.println(toString());
}
```

Assume further that the method **showAdvertisement()** is not overridden in the class **DiscountSale**. What is the output of the following code?

```
Sale s = new Sale("floor mat",10.00);
DiscountSale discount = new DiscountSale("floor mat",11.00,10);
s.showAdvertisement();
discount.showAdvertisement();
```

Question 3 [Polymorphism & Inner Classes] (3*5 = 15 Marks)

Consider a class called **Employee** with the following fields:

- name (**String** type)
- hired (**Date** type)
- salary (**double** type)

1. Provide the following:

   a) A constructor that initializes the three fields (**name**, **hired** and **salary**). Provide all the accessors and mutators for the three fields.

b) A **toString()** method that returns a **String** object containing the name of the employee, his date of hiring and his salary. It should be noted that the class **Date** has its own implementation of the **toString()** method.

2. The class **Employee** overrides the **equals()** method. Provide an implementation that returns **true** if two **Employee** objects have the same name, the same date of hiring, and the same salary and **false** otherwise. Note that the class **Object** defines the **equals()** method using the following signature: **public boolean equals(Object o)**

3. Write an inner class called **Retreat** that has as members: retreat date, retreat salary and a method which has the same name as the salary accessor method in the outer class and which computes the retreat salary as follows:
   - 30% of the employee current salary if the difference between the retreat date year and the hiring date year is strictly less than 20 years.
   - 50% of the employee current salary if the difference between the retreat date year and the hiring date year is more than or equal 20 years and strictly less than 30.
   - 80% of the employee current salary if the difference between the retreat date year and the hiring date year is more than or equal 30 years.

   Hint: Think of using the method **getYear()** in the class **Date** to extract the year of a specific date.

Question 4 [JVM] (2*5 = 10 Marks)

(a) What are the three sub processes in linking in JVM?

(b) Explain, with the help of an example, one method in the class **java.lang.Class**.

Question 5 [Exceptions] (15 Marks)

Write a Java program that calculates the average of *N* integers. The program should prompt the user to enter the value for *N* and then afterward must enter all *N* numbers. If the user enters a non-positive value for *N*, then an exception should be thrown and caught with the message “N must be positive”. If there is any exception as the user is entering the *N* numbers, an error message should be displayed and the user prompted to enter the number again.

Question 6 [Recursion] (20 Marks)

One way for finding the maximum of an array of integers is to divide the array into two halves, find the maximum m_1 in the first half and m_2 in the second half through recursive calls, and then return the larger of m_1 and m_2. Write the method in Java using the following heading: **public static int findMax(int[] a, int start, int end)**