بسم الله الرحمن الرحيم

King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Information and Computer Science Department
Fall Semester (071)
ICS 102 - Introduction to Computing I

Major Exam 02

Name:

ID#:

Please circle your section number below:

Question #	Maximum Marks	Obtained Marks
1	20	
2	40	
3	10	
4	15	
5	15	
Total	**100**	

~Good Luck ~

Q1. **[4 * 5 = 20 marks]** Consider the following class:

```
class Circle {

      private double radius;

      Circle() {
            radius = 1;
      }

      Circle( double pradius) {
            radius = pradius;
      }

      public void set(double pradius) {
            radius = pradius;
      }

      public double get() {
            return radius;
      }

      public double area() {
            return Math.PI * radius * radius;
      }

      public boolean equals( Circle c) {
            return c.radius == radius;
      }

      public String toString() {
            return radius + "  ";
      }
}
```

Assume that I have the following statements in a *driver class* that uses the **`Circle`** class.

```
Circle c1, c2;
c1 = new Circle(5);
c2 = new Circle();
```

a) What is the value of the instance variable radius for **`c1`** and **`c2`**?

5 1

b) What does the following statement do?

```
c2.set(5);
```

assigns 5 to the instance variable "radius" of c2.

c) What is the value of **`v1`** as a result of the following statement:

```
double v1 = c1.get();
```

5

d) What is the value of **b1** as a result of the following statement:

```
boolean b1 = c1.equals(c2);
```

false

e) Is it legal to add the following method to the above class?

```
public void set() {
      radius = 5;
}
```

Yes. It overload the set method.

f) Is it legal to use the following statements?

```
c1.set(5);
c2.set(5);
boolean b1 = c1 == c2;
```

Yes. It compares the references stored in c1 and c2, therefore the value of b1 will be false

---

Q2. **[8 * 5 = 40 marks]** Give output for each of the following code in the space provided:

Code	Output

`public class M2Q1 {`
`  public static void main(String [] args) {`
`  int x = 5;`
`  int y = 3;`
`  m1(x, y);`
`  System.out.println(x + "  " + y);`
`  }`
 	
`  public static void m1(int a, int b) {`
`  int x = a*2;`
`  int y = b*2;`
`  }`
`}`
	5 3

Code	Output

`public class M2Q2 {`
`  public static void main(String [] args) {`
`  int x = 4;`
`  System.out.println(x);`
`  System.out.println(f(g(x)));`
`  System.out.println(x);`
`  }`
 	
`  public static int f(int x) {`
`  return x*x;`
`  }`
 	
`  public static int g(int x) {`
`  x = x * 10;`
`  return x;`
`  }`
`}`
	4
	4
	9

`public class M2Q3 {`
`  public static void main(String [] args) {`
`  int x = 9;`
`  Test t = new Test();`
`  t.m1();`
`  System.out.println(x);`
`  }`
`}`
`public class Test {`
`  private int x;`
`  Test() {`
`  x = 4;`
`  }`
`  public void m1() {`
`  m2(x);`
`  System.out.println(x);`
`  }`
`  public void m2(int y) {`
`  int x = y * 10;`
`  System.out.println(y);`
`  }`
`}`
	4
	1600
	4

Code	Output

```
public class M2Q4 {
   public static void main(String[] args) {
       Test t1 = new Test();
       Test t2 = new Test(3);
       Test t3 = t1;
       System.out.println(t1.getX());
       t3.setX(44);
       System.out.println(t1.equals(t3));
       System.out.println(t1);
       System.out.println(t2);
   }
}
public class Test {
   private int x, y;
   Test() {
       x = 1;
       y = 1;
   }
   Test(int p) {
       x = p;
       y = p;
   }
   public void setX(int p) {
       x = 44;
   }
   public int getX() {
       return x;
   }
   public boolean equals(Test t) {
       return t.x == x && t.y == y;
   }
   public String toString() {
       return x + " " + y;
   }
}
```

```
1
true
44 1
3 3
```

Q3. **[10 marks]** Write a Class Student that has:

- The Student Name
- His ID, where ID must be 6 digits (i.e. > 99999 and < 1000000)
- An accessor (setter) for each variable.
- A mutator (getter) for each variable.
- equals method (comparing the IDs)

```
class Student {

        String name;
        int ID;

        public String getName()
                { return name; }

        public int GetID()
                { return ID; }

        public boolean  setID( int pId) {
                if(pId < 99999 || pId > 1000000) return false;
                ID = pId;
                return true; }

        public Void setName(String pname)
                {name = pname; }

        public Boolean equals(Student s)
                { return ID == s.ID; }
}
```

Q4. **[15 marks]** Write a Class Ellipse that has

- long and short axes *a* and *b* as instance variables
- A constructor with both axes,
- Another constructor with No parameters (both axes are set to1)
- **area** and **perimeter** methods.
- **toString** method that return both axes and the area.

Note: area = πab and perimeter $\approx 2\pi\sqrt{\frac{1}{2}(a^2+b^2)}$

```
public class Ellipse {

    int a,b; // it can be double too

    public Ellipse(int a, int b)
    {
        this.a = a;
        this.b = b;
    }

    public Ellipse()
    {
        a = b = 1; //or this(1,1)
    }

    public double area()
    {
        return Math.PI * a * b;
    }

    public double perimeter()
    {
        double inSqrt =  1.0/2 * ( Math.pow(a,2) + Math.pow(b,2) );
        return 2 * Math.PI * Math.sqrt( inSqrt );
    }

    public String toString()
    {
        return "a = " + a + ", b = " + b + ", area = " + area();
    }
```

Q5. **[15 marks]** Fill in the blanks following the comments (the blank might be empty!)

```
public class Point {
    private int x, y;

    public Point(int x, int y){
        this.x = x;
        this.y = y;
    }
    public Point(){
        x = y = 0;
    }
    public Point(int x){
        this(x,0);
    }

    public int getX() { return x; }
    public int getY() { return y; }

    public void setX(int x) { this.x = x; }
    public void setY(int y) { this.y = y; }

    public void move(int deltaX, int deltaY){
        x += deltaX;
        y += deltaY;
    }
// complete the equals method below…

    public _boolean__ equals(_Point p____){


        return ___x == p.x && y == p.y____  ;

    }
// complete the toString method below… to return "[ x ; y ]"

    public _String___ toString(___---____){


        return _"[ " + x + " ; " + y + " ]"_;

    }
}
public class M2Q2 {
    public static void main(String [] args) {

    // Create two points p and q
    Point p = new Point();
    Point q = new Point(7,3);

    // call move method to p with 2 and 5
    p.move(2,5);

    // print the toString of q
    System.out.println(q);

    // test if p equals q and print the result.
    System.out.println(p.equals(q));

    }
}
```