Question one (25%)

Part1_Multiple choice (15%)

1. **The expression (int)(76.0252175 * 100) / 100 evaluates to _____.**

   (a.) 76 b. 76.0252175 c. 76.03 d. 76.02

2. **Given the declaration Circle x = new Circle (), which of the following statement is most accurate.**

   a. x contains an object of the Circle type.
   b. You can assign an int value to x.
   (c.) x contains a reference to a Circle object.
   d. x contains an int value.

3. **To declare a constant MAX_LENGTH inside a method with value 99.98, you write**

   a. double MAX_LENGTH = 99.98;
   b. final MAX_LENGTH = 99.98;
   (c.) final double MAX_LENGTH = 99.98;
   d. None of the above.

4. **Which of the following expression yields an integer between 0 and 100, inclusive?**

   a. (int)(Math.random() * 100 + 1)
   (b.) (int)(Math.random() * 101)
   c. (int)(Math.random() * 100)
   d. (int)(Math.random() * 100) + 1

5. **Analyze the following code:**

```
class Test {
    public static void main(String[] args) {
        System.out.println(xMethod((double)5));
    }
```

```
    public static int xMethod(int n) {

        System.out.println("int");
        return n;
    }
    public static long xMethod(long n) {
        System.out.println("long");
         return n;
    }
}
```

a. The program displays int followed by 5.

b. The program displays long followed by 5.

c. The program runs fine but displays things other than 5.

(d.) The program does not compile.

Answer sheet

1	a
2	c
3	c
4	b
5	d

3 Mark each

Part 2(10%)

Is the following class immutable? If the answer is NO, do any necessary modification(s) to make the class immutable.

```
class  A {
    private int x;
    public double y;
    private int [] values;
    public void setX(int x){
          this.x = x;     }
    public int getX(){
        return x;
    }
    public int [] getValues(){
        return values;
    }
}
```

No (1 Mark)

3 Mark

4 Mark

```
class A {
private int x;
private double y;
private int[] values;
public int getX() {
return x;
}
```

3 Mark

Question Two (20%)-Java Language

1. **Find the two errors of the following Java code(assume that class circle is given)**

```
public class Foo{
      public static void main(String []args){
          method1();
      }

      public void method1(){
          method2();
      }

      public static void method2(){
          System.out.println("what is radius "+
                             c.getRadius());
      }

      Circle c = new Circle();
}
```

(2.5)

(2.5)

1) method1 is an instance member cannot be called From a static member

2) C is an instance object cannot be accessed in a static member

3) new is invoked in a wrong place

2. **What is wrong in the following code?**

```
class Test {
     public static void main(String[] args) {
        A a = new A();
        a.print();
     }
}

class  A {
     String s;
     A(String s) {
        this.s = s;
     }
     public void print() {
        System.out.print(s);
     }
}
```

No default Constructor

ambigious code

5 Mark

3. **What is the printout of the following code?**

```
public class Foo {
    private boolean x;
    public static void main(String[] args) {
        Foo foo = new Foo();
        System.out.println(foo.x);
    }
}
```

Output:

5 Mark

False

4. **Show the output of the following program:**

```
public class Test {
  public static void main(String[] args) {
    Count myCount = new Count();
    int times = 0;

    for (int i = 0; i < 100; i++)
      increment(myCount, times);

    System.out.println("count is " + myCount.count);
    System.out.println("times is " + times);
  }

  public static void increment(Count c, int times) {
    c.count++;
    times++;
  }
}
```

```
class Count {
  public int count;

  Count(int c) {
    count = c;
  }

  Count() {
    count = 1;
  }
}
```

output:

5 Mark

Count is 101

times is 0

Question Three (20%)-Java Language

Given the following UML

Fan
-speed:int -on:boolean -radius:double
+Fan(s:int,o:boolean,r:double) +toString():String

1) Write the code of the class Fan and notice that toString will return a string description whether the fan is on or off.

(7) 8

```
class Fan {
Private int speed;
Private boolean on;
Private double radius;
Public Fan(int s, boolean o, double r){
        speed = s;
        on = o;
    } radius = r;
Public String toString() { return on;}
```

2) Write the statement to instantiate a Fan object with a speed of 6, the fan is on and radius of 5.0.

4 (5)

```
Fan myFan = new Fan(6, true, 5.0);
```

3) Write a statement to invoke the toString method for previous created object.

4 (5)

```
myFan.toString();
```

4) Add a constructor to class Fan that only initiates the fan status whether it's off or on.

4 (3)

```
public Fan(boolean on){
    this.on = on;
}
```

Question Four (35%) - Design and Programming

Write a class to represent an Account description. It should have as attributes the account id, balance and description each of which can be represented as integer, string and double. The member methods of the class should include.

- Default constructor.
- Setter and getter methods for all members assuming that id will be unchanged after instance creation.
- Deposit method (خدمة الإيداع)
- Withdraw method(خدمة سحب)
- A non-member method to print all the details of Account.

Write a driver program to test the class Account. By keeping track of 3 different customer accounts and do the following.

The first account contains 1000, second 2000 and third contains 3000.

Try to deposit the first account and second amount by 1000.

Withdraw the third account by 3500.

Finally print the details of all accounts before the changes and after the changes.

(You should do all the necessary validation(s) in withdraw method).

```
Class Account {
~~Public~~ Private int id;
        Private double balance;
        Private String desc;
        public static int counter = 0;
Public Account () {
  id = Counter;
  balance = 0;
  desc = "normal";
}
```

6 Mark

setters and getters for all instance members except SetId ⟶ 2 Marks for those who put this method.

```
Public void deposit (double amount) {
    balance += amount;
}
```

5 Marks

```
Public void withdraw (double amount) {
    if (amount < balance)
        balance -= amount;
    else
        System.out.println ("Cannot withdraw You don't
                             have enough money");
    }
}
```

7 Marks

```
Public class Test {
    Public static void main (String [] args) {
        Account [] array = new Account [3];
        for (int i=0; i<3; i++)
            array [i] = new Account ();
        array [0]. setBalance (1000);
        array [1]. setBalance (2000);
        array [2]. setBalance (3000);
        for (int i=0; i<3; i++)
            print (array [i]);
        array [0]. deposit (1000);
        array [1]. deposit (1000);
        array [2]. withdraw (500);
        for (i=0; i<3; i++)
            print (array [i]);
    }
```

7 Marks

3 Marks

```
Public static void
    print (Account C) {
System.out.println (
    C.getId() + C.getBa...
        + C.getDes();
}
```