Mid Term Exam – Fall 2020

Course code: CSE215 **Total Mark: 50** **Time: 60 Min** **Date: 10 December 2020**

Stack
- top : int - arr[] : int
+ Stack() + Stack(a : int) + push(a : int): void + pop() : int + getStackSize() : int + displayStack(): void

```
public class TestStack {
        public static void main (String[] abc)
        {
                Stack s = new Stack(5);
                s.push(1); s.push(5); s.push(8); s.push(80); s.push(2);
                s.pop();
                s.pop();
                s.displayStack();
                System.out.println("\n" + "The size of the stack: " +
s.getStackSize());
        }
}
/* This code should yield the following output:
1 5 8
The size of the stack: 3 */
```

1. **(20 points)** Implement the class Stack

- Data field 'top' holds the index of the last element of the data field arr[].arr[] is the storage of the stack. Initially, as there will be no elements in the stack so top should have an initial value -1. Integer elements can be inserted and popped out from arr with the last in first out (LIFO) manner.
- Stack() is a no-argument constructor that defines the arr[] with a default size 10 and initialize it with all zeros. Stack(a : int) is another constructor that defines the arr[] with size a and also initialize it with all zeros.
- push(a : int) inserts the value in a to the top position of arr[]
- pop() returns the value of the top position of arr[] and removes that value from arr[]
- getStackSize() returns the number of elements currently holding inside arr
- displayStack() prints currently holding elements of the array.

2. **(5 points)** "Encapsulation keeps the data and codes safe from external inheritance." - Explain this statement with a code example.

3. **(5 points)** Write a method to randomly initialize a 2D array of size NxN with the constrain that the main diagonal locations of the array are strictly zeros.

4. **(5 points)** Explain the errors in the following program.

```
public class Apple extends Fruit {
  Apple a = new Apple();
}
class Fruit {
  public Fruit(String name) {
          System.out.println("Fruit's constructor is invoked");
          }
}
```

5**. (1+2+2 points)** What does the immutable object mean? The instance of which classes are immutable? Describe the object's immutability with a suitable code example.

6. **(2+3 points)** Print and explain the output of the following programs.

```
public class ClassA {
    int a;
    ClassA()
    {
        System.out.println(++a);
    }
    public ClassA(int b)
    {
        System.out.println(a+=b);
    }
}
```

```
public class ClassB {
    public static void main(String[] abc)
    {
        ClassA a = new ClassA();
        ClassA b = new ClassA(3);
    }
}
```

7. **(5 points)** Find and explain the problems of the following code:

```
1  public class ClassA {
2      final static int a;
3      int b;
4      static void f1()
5      {
6          b = a;
7          f2();
8      }
9      void f2()
10     {
11         int c = a;
12         public char c = 'a';
13         static double v;
14         a = b;
15     }
16 }
```