Lab Assignment 02

HOMEWORK

Task 1

Design the “**ImaginaryNumber**” class to generate the **output** given below:

Tester Class	Output

public class Tester7{
  public static void main(String [] args){
     ImaginaryNumber num1 = new ImaginaryNumber();
     String p = num1.printNumber();
     System.out.println(p);
     System.out.println("1********");
     num1.realPart=3;
     num1.imaginaryPart=7;
     System.out.println(num1.printNumber());
     System.out.println("2********");
     ImaginaryNumber num2 = new ImaginaryNumber();
     num2.realPart=1;
     num2.imaginaryPart=9;
     System.out.println(num2.printNumber());
    }
}
	0 + 0i
	1********
	3 + 7i
	2********
	1 + 9i

Task 2

Implement the “**Assignment**” class with necessary properties, so that the given output is produced for the provided driver code.

Driver Class	Output

public class AssignmentTester{
  public static void main(String [] args){
    Assignment as1 = new Assignment();
	Number of tasks: 0
	Difficulty level: null
	Submission required: false

```
    as1.printDetails();
    System.out.println("1---------------");
    as1.tasks = 11;
    as1.difficulty = "Moderate";
    as1.submission = true;
    as1.printDetails();
    System.out.println("2---------------");
    System.out.println(as1.makeOptional());
    System.out.println("3---------------");
    as1.printDetails();
    System.out.println("4---------------");
    Assignment as2 = new Assignment();
    as2.tasks = 12;
    as2.difficulty = "Hard";
    as2.submission = false;
    as2.printDetails();
    System.out.println("5---------------");
    System.out.println(as2.makeOptional());
  }
}
```

```
1---------------
Number of tasks: 11
Difficulty level: Moderate
Submission required: true
2---------------
Assignment will not require submission
3---------------
Number of tasks: 11
Difficulty level: Moderate
Submission required: false
4---------------
Number of tasks: 12
Difficulty level: Hard
Submission required: false
5---------------
Submission is already not required
```

Task 3

Create an **Employee** class to provide the expected output.

- An employee will have a name, salary and designation.
- The name will be assigned inside the newEmployee() method
- Whenever a New Employee joins his/her salary will be **Tk. 30,000** and the designation will be **junior**.
- Employees with salaries greater than **Tk. 50,000** and **Tk. 30,000** need to pay **30%** and **10%** of salary as tax respectively.
- Employees can be promoted to **senior**, **lead** and **manager** positions. Based on their promotion they will get an increment of **Tk. 25,000**, **Tk. 50,000** and **Tk. 75,000** respectively.

Driver Code

```java
public class Tester9{
  public static void main(String[] args){

    Employee emp1 = new Employee();
    Employee emp2 = new Employee();
    Employee emp3 = new Employee();

    emp1.newEmployee("Harry Potter");
    emp2.newEmployee("Hermione Granger");
    emp3.newEmployee("Ron Weasley");
    System.out.println("1 ==========");
    emp1.displayInfo();
    System.out.println("2 ==========");
    emp2.displayInfo();
    System.out.println("3 ==========");
    emp3.displayInfo();
    System.out.println("4 ==========");
    emp1.calculateTax();
    System.out.println("5 ==========");
    emp1.promoteEmployee("lead");
    System.out.println("6 ==========");
    emp1.calculateTax();
    System.out.println("7 ==========");
    emp1.displayInfo();
    System.out.println("8 ==========");
    emp3.promoteEmployee("manager");
    System.out.println("9 ==========");
    emp3.calculateTax();
    System.out.println("10 ==========");
    emp3.displayInfo();
  }
}
```

Expected Output

```
1 ==========
Employee Name: Harry Potter
Employee Salary: 30000.0 Tk
Employee Designation: junior
2 ==========
Employee Name: Hermione Granger
Employee Salary: 30000.0 Tk
Employee Designation: junior
3 ==========
Employee Name: Ron Weasley
Employee Salary: 30000.0 Tk
Employee Designation: junior
4 ==========
No need to pay tax
5 ==========
Harry Potter has been promoted to lead
New Salary: 80000.00 Tk
6 ==========
Harry Potter Tax Amount: 24000.0 Tk
7 ==========
Employee Name: Harry Potter
Employee Salary: 80000.0 Tk
Employee Designation: lead
8 ==========
Ron Weasley has been promoted to manager
New Salary: 105000.00 Tk
9 ==========
Ron Weasley Tax Amount: 31500.0 Tk
10 ==========
Employee Name: Ron Weasley
Employee Salary: 105000.0 Tk
Employee Designation: manager
```

Task 4

Implement the “**MobilePhone**” class with necessary properties to produce the given output for the provided driver code.

Driver Class

```
public class MobilePhoneTester{
  public static void main(String args []){
    MobilePhone m1 = new MobilePhone();
    MobilePhone m2 = new MobilePhone();
    m1.setContactCapacity(5);
    m2.setContactCapacity(100);
    m1.details();
    System.out.println("1----------------");
    m1.addContact("John", 9866);
    m1.addContact("Maria", 7865);
    System.out.println("2----------------");
    m1.details();
    System.out.println("3----------------");
    m1.makeCall(9866);
    System.out.println("4----------------");
    m1.addContact("Henry", 2365);
    System.out.println("5----------------");
    m1.makeCall(7552);
    m1.makeCall(2365);
    System.out.println("6----------------");
    m1.addContact("Gomes", 4589);
    m1.addContact("Antony", 8421);
    m1.addContact("Tony", 5789);
    System.out.println("7----------------");
    m1.details();
  }
}
```

Output

```
Total Contacts: 0
Contact List:
1----------------
The contact of John is added.
The contact of Maria is added.
2----------------
Total Contacts: 2
Contact List:
John:9866
Maria:7865
3----------------
Calling John . . .
4----------------
The contact of Henry is added.
5----------------
Calling 7552 . . .
Calling Henry . . .
6----------------
The contact of Gomes is added.
The contact of Antony is added.
Storage Full!!
7----------------
Total Contacts: 5
Contact List:
John:9866
Maria:7865
Henry:2365
Gomes:4589
Antony:8421
```

Task 5

```
public class B {
   public int temp = 4;
   public int sum, y, x;
   public void methodA(int m){
       int [] n = {2,5};
       int x = 0;
       y = m + this.methodB(x++,m)+(temp++);
       x = this.x + 2 + n[0];
       sum = sum + x + y;
       n[0] = sum + 2;
       System.out.println(n[0]+" " + x+ " " + sum);
   }
   public int methodB(int m, int n){
       int y = 4 + this.y + m;
       x = this.y + y + (++temp) - n;
       sum = x + y + this.sum;
       System.out.println(y+ " " + this.x + " " +sum);
       return x;
  }
}
```

```
public class Tester11 {
   public static void main(String [] args){
       B t1 = new B();
       t1.methodA(5);
       B t2 = new B();
       t2.methodB(12, 2);
   }
}
```

Outputs		

Task 6

```
public class A{
  public int x = 3, y = 5, sum = 9;
  public int methodA(int temp, int x){
    this.x += (x++) + temp;
    if(temp % 5 == 2){
      sum += (this.y++) + temp;
    }
    else{
      sum += 3;
      if(y > 5) ++y;
    }
    System.out.println(this.x + " " + y + " " + sum);
    return this.x;
  }
  public void methodB(int y){
    int temp = (y++) + this.y;
    this.y = (++temp) + methodA(temp, y) + x;
    sum = y + (++this.x) + (temp++);
    System.out.println(x + " " + y + " " + (sum++));
  }
  public void methodC(int y){
    y = (this.x++) + sum + 3;
    System.out.println(x + " " + y + " " + sum);
  }
}
```

Driver code:

```
public class Test12{
  public static void main(String [] args) {
   A a1 = new A();
   a1.methodA(2, 4);
   a1.methodB(3);
   new A().methodC(7);
  }
}
```

Outputs		