Lab Assignment 04

HOMEWORK

Task 1

Design the **Player** class to obtain the following output from the driver code. You can assume, each player can defeat at most 5 villains.

Driver Code

```
public class PlayerTester {
  public static void main(String[] args) {
    Player ben = new Player("Ben", 10);
    System.out.println("======1======");
    ben.viewInfo();
    System.out.println("======2======");
    ben.defeatVillain("Vilgax", 100);
    System.out.println("======3======");
    ben.defeatVillain("Yamcha", 10);
    System.out.println("======4======");
    ben.viewInfo();
    System.out.println("======5======");
    ben.defeatVillain('8', "Vilgax", 100);
    System.out.println("======6======");
    ben.viewInfo();
    System.out.println("======7======");
    Player kevin = new Player("Kevin");
    System.out.println("======8======");
    kevin.viewInfo();
    System.out.println("======9======");
    ben.defeatVillain(kevin);
    System.out.println("======10======");
    ben.viewInfo();
    System.out.println("======11======");
    Player goku = new Player("Goku", 9000);
    System.out.println("======12======");
    ben.defeatVillain(goku);
  }
}
```

Expected Output

```
Ben joined the game
HP: 10
======1======
Player Name: Ben
Current HP: 10
======2======
failed to defeat Vilgax
======3======
defeated Yamcha
======4======
Player Name: Ben
Current HP: 20
Defeated:
Yamcha,
======5======
HP with 8x boost: 160
defeated Vilgax
======6======
Player Name: Ben
Current HP: 260
Defeated:
Yamcha, Vilgax,
======7======
Kevin joined the game
HP: 100
======8======
Player Name: Kevin
Current HP: 100
======9======
defeated Kevin
======10======
Player Name: Ben
Current HP: 360
Defeated:
Yamcha, Vilgax, Kevin,
======11======
Goku joined the game
HP: 9000
======12======
failed to defeat Goku
```

Task 2

Write “**Student**“ class to show the following expected outputs

Note:

- A student can’t take any course until the CGPA is set.
- A student cannot take more than 4 courses.
- A student with CGPA below 3 cannot take more than 3 courses.

Driver Code

```
public class StudentDriver {
  public static void main(String[] args){
    Student student1 = new Student(12345678);
    System.out.println("1-----------------");
    student1.addCourse("CSE110");
    System.out.println("2-----------------");
    student1.storeCG(2.5);
    student1.addCourse("CSE110");
    student1.addCourse("ENG101");
    student1.showAdvisee();
    System.out.println("3-----------------");
    student1.removeAllCourse();
    student1.showAdvisee();
    System.out.println("4-----------------");
    student1.storeID(54652365);
    String[] courses = {"SOC101","CSE111","ENG102"};
    student1.addCourse(courses);
    student1.showAdvisee();
    System.out.println("5-----------------");
    student1.addCourse("CSE230");
    student1.showAdvisee();
    System.out.println("6-----------------");
    Student student2 = new Student(975738383,3.7);
    System.out.println("7-----------------");
    String[] courses2 =
{"CSE220","PHY112","MAT120","BUS101","CHN101"};
    student2.addCourse(courses2);
    student2.showAdvisee();
  }
}
```

Expected Output

```
A student with ID 12345678 has
been created.
1-----------------
Failed to add CSE110
Set CG first
2-----------------
Student ID: 12345678, CGPA: 2.5
Added courses are:
CSE110 ENG101
3-----------------
Student ID: 12345678, CGPA: 2.5
No courses added.
4-----------------
Student ID: 54652365, CGPA: 2.5
Added courses are:
SOC101 CSE111 ENG102
5-----------------
Failed to add CSE230
CG is low. Can't add more than 3
courses.
Student ID: 54652365, CGPA: 2.5
Added courses are:
SOC101 CSE111 ENG102
6-----------------
A student with ID 975738383 and
cgpa 3.7 has been created.
7-----------------
Failed to add CHN101
Maximum 4 courses allowed.
Student ID: 975738383, CGPA: 3.7
Added courses are:
CSE220 PHY112 MAT120 BUS101
```

Task 3

Design the **Student** and the **Connect** class so that the following output is produced.
Note:

- A student's email, password, and login status are null by default while creating an object of the Student class.
- Your code should satisfy the conditions mentioned in the output only.
- Connect class will have two instance variables: totalAdvisee and an array of Student type to store the student object. The array will be updated inside the advising() method only when the advising is successful.
- Connect can take at most 5 advisees.

<table>
<tr><th>Driver Code</th><th>Expected Output</th></tr>
<tr><td>

```
public class ConnectTester {
  public static void main(String[] args) {
    Student rakib = new Student("Rakib", 12301455, "CSE");
    Student roy = new Student("Roy", 12501345, "CS");
    System.out.println("1*************");
    Connect connectObj = new Connect();
    System.out.println("2*************");
    connectObj.login(rakib);
    System.out.println("3*************");
    connectObj.advising(rakib);
    System.out.println("4*************");
    rakib.email = "rakib@hotmail.com";
    rakib.password = "1234";
    System.out.println("5*************");
    connectObj.login(rakib);
    System.out.println("6*************");
    connectObj.advising(rakib);
    System.out.println("7*************");
    connectObj.advising(rakib, "CSE110", "PHY111", "MAT110",
"CSE260");
    System.out.println("8*************");
    connectObj.advising(rakib, "CSE110", "PHY111","MAT110");
    System.out.println("9*************");
    connectObj.allAdviseeInfo();
    System.out.println("10*************");
    roy.email = "roy@hotmail.com";
    roy.password = "abcd";
    connectObj.login(roy);
    System.out.println("11*************");
    connectObj.advising(roy, "CSE110", "ENG101", "PHY112");
    System.out.println("12*************");
    connectObj.allAdviseeInfo();
  }
}
```

</td><td>

```
Student object is created
Student object is created
1*************
Connect is ready to use!
2*************
Email and password need to be set.
3*************
Please login to advise courses!
4*************
5*************
Login successful
6*************
You haven't selected any courses.
7*************
You need special approval to take
more than 3 courses.
8*************
Advising successful!
9*************
Total Advisee: 1
Name: Rakib ID: 12301455
Department: CSE
Advised Courses:
CSE110 PHY111 MAT110
==============
10*************
Login successful
11*************
Advising successful!
12*************
Total Advisee: 2
Name: Rakib ID: 12301455
Department: CSE
Advised Courses:
CSE110 PHY111 MAT110
==============
Name: Roy ID: 12501345
Department: CS
Advised Courses:
CSE110 ENG101 PHY112
==============
```

</td></tr>
</table>

Task 4

Design the **TravelPrep** class so that the given output is generated.

- This class keeps track of the package info and the destination that goes into it.
- The initial budget will be 1250 yen for any traveller.
- A traveller can visit at most 3 destinations if that falls under their budget.

Driver Code	**Expected Output**

public class TravelPlan {
  public static void main(String[] args) {
    TravelPrep t1 = new TravelPrep();
    System.out.println("1==================");
    TravelPrep t2 = new TravelPrep("Package-2", 2100);
    System.out.println("2==================");
    TravelPrep d1 = new TravelPrep("Fushimi", "Shrine");
    TravelPrep d2 = new TravelPrep("Lake", "Kawaguchi", 550);
    TravelPrep d3 = new TravelPrep("Shrine", "Hieizan", 1000);
    TravelPrep d4 = new TravelPrep("Lake", "Ashi", 620);
    System.out.println("3==================");
    t2.add_to_itinerary(d1);
    t2.add_to_itinerary(d2, d3);
    t2.show_itinerary();
    System.out.println("4==================");
    System.out.println(d2.updateCost(60));
    System.out.println("5==================");
    t1.t_name = "Package-1";
    t1.add_to_itinerary(d2, d4);
    t1.add_to_itinerary(d3);
    System.out.println("6==================");
    t1.show_itinerary();
  }
}
	Unknown package would cost at most 1250 yen
	1==================
	Package-2 would cost at most 2100 yen
	2==================
	Fushimi Shrine costs 300 yen
	Kawaguchi Lake costs 550 yen
	Hieizan Shrine costs 1000 yen
	Ashi Lake costs 620 yen
	3==================
	Itinerary for Package-2
	1. Fushimi Shrine - 300 yen
	2. Kawaguchi Lake - 550 yen
	3. Hieizan Shrine - 1000 yen
	Total cost of 3 destinations: 1850 yen
	4==================
	Cost of Kawaguchi Lake updated to 610
	5==================
	Budget going overboard
	6==================
	Itinerary for Package-1
	1. Kawaguchi Lake - 610 yen
	2. Ashi Lake - 620 yen
	Total cost of 2 destinations: 1230 yen

Task 5

```
public class TestX{
 public int result = 17, i, n = 7, m = 5;
 public int[] num={6,8,11};
 public void compute(int[] nums){
   n = nums[0] + this.m;
   this.result += nums[i];
   TestX temp = new TestX();
   boolean check = !(temp.modify(temp, num));
   if (check) {
     m = nums[1] + temp.result;
   } else {
     m = result - nums[3];
   }
   System.out.println(m + " " + n + " " + result);
 }
 public boolean modify(TestX obj, int[] arr){
   arr[1]++;
   i=i+1;
   obj.m = this.result + arr[i];
   obj.result = obj.m - obj.n + n;
   System.out.println(obj.m + " " + obj.n + " " + result);
   return arr[1] % 2 == 0;
 }
 public boolean modify(TestX obj){
   num[i]++;
   i=i+1;
   obj.m = this.result + num[i];
   obj.result = num[2] - obj.n + n;
   System.out.println(obj.m + " " + obj.n + " " + result);
   return num[1] % 2 != 0;
 }
}
```

```
public class tester11{
  public static void main(String[] args){
    TestX tx = new TestX();
    int[] data = {4, 6, 2, 3, 5};
    tx.compute(data);
    tx.modify(tx);
  }
}
```

Outputs		

Task 6

```java
public class A {
 public int x, y = 2, sum = 5;
 public A(){
   x = sum - y;
   this.methodB(1);
 }
 public A(int x){
   this.x = x;
 }
 public void methodB(int y) {
   int temp = (y++) + (++this.y);
   sum = y + temp - ++this.x;
   System.out.println(y + " " + temp + " " + (sum++));
   this.y = (++temp) + methodC(temp, y) + x;
 }
 public void methodA(int temp, int x) {
   A a2 = new A(6);
   a2.sum = this.methodC(a2);
   System.out.println(this.x + " " + a2.sum + " " + y);
   this.sum = a2.methodC(this);
   System.out.println(a2.x + " " + this.sum + " " + a2.y);
 }
 public int methodC(A a) {
   return this.x + a.y + a.sum;
 }
 public int methodC(int sum, int y) {
   y = (this.x++) + sum + 3;
   return y;
 }
}
```

```java
public class Tester12{
  public static void main(String [] args) {
    A a1 = new A();
    a1.methodA(4, 3);
  }
}
```

Outputs		