SLIIT
Discover Your Future

BSc (Hons) in Information Technology
Year 2

Tutorial 3

IT2030 – Object Oriented Programming **Semester 1, 2023**

Exercise 1

a) Create a class called `Student` with the properties studentID(int), name(string), degree (string), mobile(string).

b) Create a constructor for the Student class to set values for name, degree and mobile number. The student id should be generated by the system making use of a static property called maxwhich is initialized to 100 initially to assign a unique value for each student id.

c) Create setters and getters for name, degree and mobile.

d) Create a **print()** method to print details of a student.

e) Create a static method called **getNextStudentID()** to return the next student id.

f) Create a separate class called `StudentApp` with the main method.

g) Create an array of Student objects. Instantiate 5 Student objects using the constructors and the setters you have developed.

h) Display the details of all the 5 students making use of the for loop.

Exercise 2

01. Implement the following class

```
class Feet {
      private int feet;
      private int inches;
      public Feet(int feet, int inches){}
      // Add f1+f2 feet and store in current feet
     public void add(Feet f1, Feet f2){}
      // Display a Length e.g 5'6"
      public void print() {}
}
```

a) Write a separate program and a main function to test the above class.

b) Overload the print() function to add a message to be printed in front of the length.

```
public void print(String msg) {…}
e.g.
      Feet mylength = new Feet(5,6);
      mylength.print("Length : "); // should print Length : 5'6"
```

c) Implement an overloaded constructor that can accept another Feet object.

```
public Feet(Feet len) {}
// Copy the content of len to the new Feet Object.
```

d) Implement an overloaded add method that adds the current length to the new length and stores it in the current Feet object.

```
public void add(Feet f1) {…}

e.g.
      Feet mylength = new Feet(5,6);
      Feet newlen = new Feet(6,7)
      mylength.add(newlen);
      mylength.print(); // 12'1"
```

e) Implement a static print method for Feet so that any Feet object can be printed using the static method.

```
public static void print(Feet f) {…}

e.g.
      Feet mylength = new Feet(5,6);
      Feet.print(mylength);
```

f) Why can't you have the following static add()method, here we want to return a Feet object.

```
public static Feet add(Feet f1, Feet f2){}
```

g) Instead implement the following static add() method which is used to add three Feet objects and return a new Feet object

```
public static Feet add(Feet f1, Feet f2, Feet f3) {}

e.g.
      Feet f5 = Feet.add(f1, f2, f3);
```

Exercise 3

Implement a class called Calculation with two static methods that calculate the addition of two numbers the subtraction of two numbers. Implement a class called DemoApp and in the main function call the two methods directly without creating objects