Software and Programming 1

Week 8 Lab -
Use of Classes
and Inheritance
Lab Objectives

Exercise 1 - Student & StudentTest classes

1. Reinforce Code Writing
   1. Class structure
   2. Accessor (getter)/ Mutator (setter) methods.
2. Override an inherited method
3. Use of array to create multiple Objects
4. Use of Enhanced Loop / String comparison

Exercise 2 - Person, Tutor, Student & Test classes

1. Class Inheritance
2. Code Revision / Refactoring
Exercises 1: Classes Student and StudentTest

Implement a class **Student**
The Class requires the following attributes:
   name, year of birth and programme of study

1. Write the Class declaration
2. Declare variables
3. Create constructor
4. Write three new methods:
   String getName(), int getYear() String getProgramme()
5. Override the method String toString() from the class Object.

Note: as there are no mutator methods, the data for instances must be supplied on creation (through Constructor).
Exercises 1: Classes Student and StudentTest (2)

Implement a test class `StudentTest`

The StudentClass is required to:

1. Create an array of 10 Students

2. Create instances and receive user input to provide data to each Student object
   Use a loop and a Scanner object.

3. Once data input is complete, print out the names of students of the programme "BSc ISM" only.
   Use the enhanced `for` loop for this.
/** Student class **/

public class Student
{
    /* private data */
    private String name;
    private int year;
    private String programme;

    /* methods (public interface) */
    public String getName()
    {
        /* To Do: write the code to return name */
    }
}
public int getYear()
{
    /* To Do: write the code to return year */
}

public String getProgramme()
{
    /* To Do: write the code to return programme */
}

/* refer to slide 12 of ‘Inheritance and Interfaces’ presentation */
public String toString()
{
    return "Student " + name + ", programme " + programme;
}
/* Constructor */

public Student(String name, int year, String course) {
    this.name = name;
    this.year = year;
    this.programme = course;
}

} // end of class Student
import java.util.Scanner;

public class StudentTest
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        /* To Do: Write more code to
         1. Declare and create an array of 10 Students.
         2. Fill up the array by creating 10 instances of
             students where the name, year of birth and
             programme of study are entered at the keyboard.
        */
/* To Do: Write more code to use an enhanced for loop to print out names of students of the programme “BSc ISM” only. */

} // end of method main

} // end of class StudentTest

Note: an example of the enhanced for loop can be found on slide 11 of the ‘Inheritance and Interfaces’ presentation in week 7.
Exercise 2: Classes Person, Tutor, Student (revised) and Test

2. Implement a class **Person** (a person has a name and a year of birth). Change the class Student so that it extends Person. Then implement another subclass **Tutor** of Person (a tutor has a salary).

For the classes, write the class declarations, the constructors and the methods toString() for all classes. Implement a program that tests these classes and methods.
Inheritance hierarchies

Person

- name: String
- year: int
- getName(): String
- getYear(): int

Student

- programme: String
- getProgramme(): String

Tutor

- salary: double
- getSalary(): double
/** Revised Student class **/  
public class Student extends Person  
{  
  private String programme;  
  public String getProgramme( )  
  {  
    /* To Do: write the code to return programme */  
  }  
  
  public String toString()  
  {  
    return "Student " + super.getName() + 
    ", programme " + programme;  
  }  
}
public Student(String name, int year, String course) {
    super(name, year); // calls the constructor of Person
    this.programme = course;
}

} // end of class Student