



Njabulo Skosana (I-LAB) Virtual Classroom

Practical Assignment 2

The java classroom is a model of a classroom that keeps a record of students and their marks and allows the educator to calculate performance indicators regarding their classroom such as the grade average number of students who passed etc.

Part 1

Within a classroom, there are people who can either be a student or a teacher. Each person has a name and surname.

1. Person class holds the name and surname of the person, this class will be the parent class for a Student and a Teacher.
2. This means Student and Teacher will inherit the name and surname variables from this class and be able to use all its getters and setters.

```
1 public class Person {
2
3     /* Do not remove the comments
4     // 1. create a string reference to the name of the person with a private access modifier eg. private String name;
5
6     // 2. create a string reference to the name of the person with a private access modifier eg. private String surname;
7
8     // 3. Add a constructor to initialize both local variables;
9
10    // 4. Add getters and setters with public access modifier for all local variables;
11
12    -----
13    Add code below
14    */
15
16    //Add all student variables with private access modifiers
17    private String name;
18    private String surname;
19 }
```

Constructors

1. A constructor is a function that will be the first to be automatically executed when a class is created.
2. A default constructor is a constructor that does not take in any arguments
3. Given the person class we can see an example of a constructor taking two arguments, the name and surname

```
20 //Add all constructors
21
22 // Default constructor
23 Person()
24 {
25     this.name="";
26     this.surname="";
27 }
28
29
30 // name and surname constructor
31 Person(String name,String surname)
32 {
33     // initialize local variables;
34     this.name=name;
35     this.surname=surname;
36 }
```

Using the `*this` keyword is a keyword used to reference the local variables within a class

Getters and Setters

- Getters are functions that return local variables of a class, usually these variables are declared private and cannot be accessed outside the class.
- Setters are functions that set the local variables of class using the argument passed in the function
- Examples of getters and setters can be seen in the Person class

```

38 //Add all getters and setters for local variables
39 //Getters
40 String getName()
41 {
42     return this.name;
43 }
44
45 String getSurname()
46 {
47     return this.surname;
48 }
49
50 //Setters
51 void setSurname(String surname)
52 {
53     this.surname=surname;
54 }
55
56 void setName(String name)
57 {
58     this.name=name;
59 }
60
61
62
63
64 }
65

```

Inheritance

Inheritance is fundamental principle in polymorphism which allows us to have classes derived from other classes and have these classes share attributes.

Within a classroom we have a teacher who is a person we can model this by having a Teacher class which will represent the teacher and a Student class which will represent a student. Both the Teacher and Student classes inherit from the person class. Every teacher has a name and a surname so does every student.

An example of this is shown with the teacher class below. Within the constructor we have to initialize the Person class by calling its constructor using the `*super` keyword. This replaces the name of the Person constructor which would be `Person(name,surname)`

```

1  public class Teacher extends Person {
2
3
4      //Add all constructors
5      Teacher(String name,String surname)
6      {
7          super(name, surname);
8      }
9
10
11 }
12

```

```

1  public class Student extends Person {
2      //Add all student variables with private access modifiers
3      private String studentNumber;
4
5      private double highestGrade;
6
7      private String highestSubject;
8
9      private double lowestGrade;
10
11     private String lowestSubject;
12
13     private Subject[] subjects;
14
15     private boolean passed;

```

Task1 [5]

Add the following variables into your student class

- The student number is a unique string that will identify a student.
- The subjects[] array is an array that holds information about subjects the student has
- The highestGrade variable holds the highest grade the student has in their subjects
- The lowestGrade variable holds the lowest grade the student has in their subjects
- The highestSubject,lowestSubject variables hold the names of the highest and lowest subject in the students array
- If the student has a mark of 50% and above for 3 or more subjects the passed variable will be true if not and the student has less than 3 subjects that are 50% and above then the passed variable holds false

Task2 [10]

- Add Getters and Setters for the Student class

Task3 [60]

- Add the following 6 functions into the Student class

```
25 //Add all helper functions for the class
26 void generateStudentNumber()
27 {
28     // Generate a unique student number a each student
29     // The student number must start with the word student
30     // then separated by a hyphen a random 8 digit number
31     // eg student-18089102
32     // then store the string in the local student number variable
33 }
34
35 void setSubjects()
36 {
37     // Generate an array of subjects where each student has 4 Subjects
38     // Subjects : Maths, LifeScience, Physical Science and English
39     // generate a random 2 digit number for each Subject and store it as the students mark for that Subject
40     // If the mark>=50 then set the Subject passed variable as true
41     // if the mark<49 then set the Subject passed variable as false
42     // then store the Subject inside the subjects array
43 }
44
45 void setHighestGradeAndSubject()
46 {
47     //Loop through the subjects array and find the highest subject with the highest grade then update the highestSubject and highestGrade variables
48 }
49
50 void setLowestGradeAndSubject()
51 {
52     //Loop through the subjects array and find the lowest subject with the lowest grade then update the lowestSubject and lowestGrade variables
53 }
54
55 void setPassed()
56 {
57     //Loop through the subjects array and if the student has passed 3 or more subjects set the passed variable to true
58     // if the student has not passed 3 or more subjects then set the passed variable to false
59 }
60 }
```

```
61 void printReportCard()
62 {
63     // print out to the terminal the report card of the student
64     /* eg.
65     * -----
66     *                               Report Card For Njabulo Skosana
67     * Student Number : student-18089102
68     * English : 50%
69     * Maths : 40%
70     * Life science : 90%
71     * Physical Science : 95%
72     * Highest Subject : Physical Science Grade: 95%
73     * Lowest Subject : Maths Grade: 40%
74     * Passed : Yes
75     */
76 }
77 }
78 }
```

Part2

1. Deep copy, when working with classes information is usually stored within the class, A deep copy is copying all attributes with a class into a new class without just assigning a reference to the class.
2. The Classroom Class will be the classroom of our program and will hold relevant information regarding a classroom

Task1 [5]

Add the following variables to the Classroom Class

```
43      //Add all student variables with private access modifiers
44      private Teacher teacher;
45
46      private Student[] students;
47
48      private Student[] topFive;
49
50      private Student[] passed;
51
52      private Student[] failed;
53
54      private double average;
55
56      private int numberOfStudents;
57
```

Task2 [15]

Add the following constructors and complete the code within the first and last constructors

1. The first constructor is default constructor if no arguments are being passed in
2. The second constructor has been completed for you as an example
3. The last constructor is a copy constructor , it has to perform a deep copy for both the Teacher and Students array.

```

60     Classroom()
61     {
62         //initialize variables
63     }
64
65     Classroom(Teacher teacher,int numberOfStudents)
66     {
67         this.teacher= new Teacher(teacher.getName(), teacher.getSurname());
68         this.students=new Student[numberOfStudents];
69     }
70
71     Classroom(Teacher teacher,Student[] students)
72     {
73         //initialize variables
74         //This function will copy all students from the given student array and store them within the classes own student array
75     }
76
77     //Add all getters and setters for local variables
78
79
80

```

Task3 [10]

1. Add all getters and setters for all local variables of the Classroom Object

Task4 [60]

2. Add all the following functions to the Classroom Class

```

81     //Add all helper functions for the class
82
83     void setStudents()
84     {
85         //This function will initialize 10 Student objects and store them in the students array
86     }
87
88     void setStudents(Student[] students)
89     {
90         //This function will copy all students from the given student array and store them within the classes own student array
91     }
92
93     void populateTop5()
94     {
95         //This function will take the top 5 students in the students array and store them into the top5 array
96     }
97
98     void populatePassed()
99     {
100         //This function will take all the students who passed in the students array and store them in the passed array
101     }
102
103     void populateFailed()
104     {
105         //This function will take all the students who failed in the students array and store them in the passed array
106     }
107
108     void calculateAverage()
109     {
110         //This function will calculate the average of the classroom eg. numberOfStudentsWhoPassed/totalNumberOfStudents
111     }
112
113
114
115
116 }

```

Due Date Is 8th 20:00 All Assignments Will Be Graded Late Submissions
Receive 0