

Object Oriented Programming

Lab 03 Access Specifiers

January 31, 2023

1 Lab Goal

The purpose of this lab is to understand how a Java class is created and what is the role of the four access specifiers on the classes, variables and methods. We will briefly explain these concepts touch up on UML and then work with a few problems. You will need to act at a brisk pace.

2 Explanation

In Java, a class can have four types of access specifiers:

- “public”: Classes, methods, and fields that are declared public can be accessed from anywhere.
- “private”: Classes, methods, and fields that are declared private can only be accessed within the same class.
- “protected”: Classes, methods, and fields that are declared protected can be accessed within the same package and by subclasses in other packages.
- “default” (or package-private): Classes, methods, and fields that do not have an access specifier are considered default, and can only be accessed within the same package. It is also worth noting that when a class is declared as private, it can’t be inherited by another class.

“public” and “private” are known as access modifiers or specifiers. They are used to indicate the accessibility of a class, method or field.

Example
- privateVar : int # protectedVar : int ~ defaultVar : int + publicVar : int
- privateMethod (): void # protectedMethod (): void ~ defaultMethod (): void + publicMethod (): void

Figure 1: Example class

See the sample code given below:

```
1 public class Example
2     { private int private
3       Var;
4       protected int protectedVar;
5       int defaultVar;
6       public int publicVar;
7
8       private void privateMethod () {
9           // private method code here
10      }
11
12      protected void protectedMethod () {
13          // protected method code here
14      }
15
16      void defaultMethod () {
17          // default method code here
18      }
19
20      public void publicMethod () {
21          // public method code here
22      }
```

3 Problems

3.1 Problem 1 - Learning Access Specifiers

Create a class called “AccessSpecifiers” with the following:

1. A private variable called “privateVar” of type int and initialize it to 0.
2. A protected variable called “protectedVar” of type int and initialize it to 0.
3. A default variable called “defaultVar” of type int and initialize it to 0.
4. A public variable called “publicVar” of type int and initialize it to 0.
5. A private method called “privateMethod” that takes no parameters and returns void. Inside the method, increment the value of “privateVar” by 1.
6. A protected method called “protectedMethod” that takes no parameters and returns void. Inside the method, increment the value of “protectedVar” by 1.
7. A default method called “defaultMethod” that takes no parameters and returns void. Inside the method, increment the value of “defaultVar” by 1.
8. A public method called “publicMethod” that takes no parameters and returns void. Inside the method, increment the value of “publicVar” by 1.
9. Create another class called “TestAccessSpecifiers” and inside the main method, create an object of the “AccessSpecifiers” class.
10. Try to access the private variable “privateVar” and the private method “privateMethod” from the “TestAccessSpecifiers” class.
11. Try to access the protected variable “protectedVar” and the protected method “protectedMethod” from the “TestAccessSpecifiers” class.
12. Try to access the default variable “defaultVar” and the default method “defaultMethod” from the “TestAccessSpecifiers” class.
13. Try to access the public variable “publicVar” and the public method “publicMethod” from the “TestAccessSpecifiers” class.
14. Run the code and observe the results.

3.2 Problem 2 - Getting student information

Create two arrays of strings where the name of the first array is first names and the name of the second array is second_names. Both the arrays will be of length 10 so assign random names to them. You will then create a **Student** class that contains the **firstName**, **lastName**, **courses** array which contains the course and the received marks as alternate entries, hence the length of the array is going to be 10. Suppose the first course is Calculus and the marks acquired were 60 and the second course was Data Structures and the marks acquired were 75, then the first four entries will be, "Calculus", "60", "Data Structures", "75" and so on.

Keep in mind that the name of the courses will be in order but the marks assigned will be randomly generated and added when the object is being created. There is going to be a private method, CalculateGrade() which will calculate the grade for every subject. Your code will create an array of 20 students using the class mentioned above and the details of every student. A sample of student details is given below.

First name: Shiraz
Second name: Farooq

Math: 60 C-
Science: 90 A-
English: 85 B
Urdu: 95 A
Chinese: 100 A+

Use any grading scheme you like. The steps that you would want to take are:

1. Create the first and last name arrays within the main method.
2. Create the array of students within the main method.
3. Create a method that returns a random name from the first and last name arrays passed to it.
4. Use that method to create random names when calling the constructor of Student class.
5. Create a method within the Student class that returns a random value between 0 and 100
6. Use this method to create the array containing courses and the marks attained
7. Create CalculateGrade() method to calculate the grades of every course
8. Create a GetDetails() method to return all the details of the Student

Keep in mind that you will need to use the correct access specifier where necessary. Failing to do so will get your marks deducted.

3.3 Problem 3 - Find the best student

In the above problem, write a function that finds the best performing student. It is possible that a student may have scored the highest marks but have actually failed a course. Therefore you need to put proper checks in place to find the best student among all the 20 students.

3.4 Problem 4 - Find the best student from a text file

Repurpose previous problems in such a way that student details are stored in the text file. After which solve Problem 3 that reads all the student data from a text file.