ICT1009 Object-oriented Programming – Lab 2

AY2021/2022

Learning outcomes:

The following lab exercises are designed to familiarize with following topics:

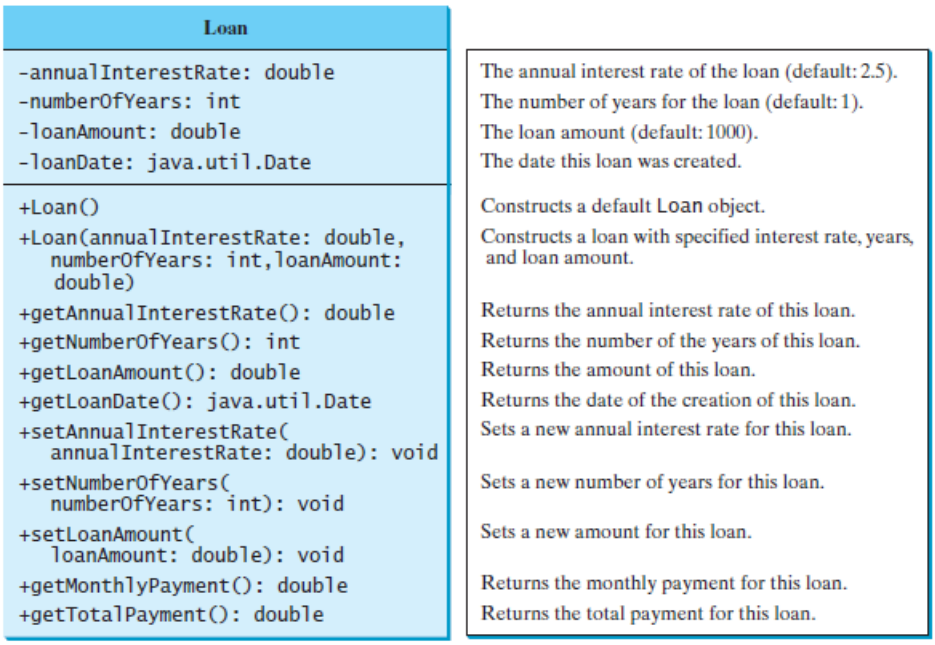
a) How to define a class
b) How to instantiate an object of the class
c) How to declare class attributes and methods
d) How to define different types of constructors for initializing object's data when the object of the class is created

Warm up exercises:

As warm up exercises, please refer the lecture notes and the attached codes and implement and try the examples discussed in the lecture related to class definition with multiple constructors & encapsulation. Please note that trying several examples in the lecture notes will give you a good warm up in picking up what was learnt in class and a good preparation for the following lab tasks.

Lab Assignment

Task 1: Consider getting a loan. A specific loan can be viewed as an object of a Loan class. The interest rate, loan amount, and loan period are its data properties, and computing the monthly payment and total payment are its methods. For example, when you buy a car, a loan object is created by instantiating the class with your loan interest rate, loan amount and loan period. You can then use the methods to find the monthly payment and total payment of your loan. Figure below indicates the Loan class UML for the Loan class.



For this task implement a Loan class with three variable attributes. The class also includes one default constructor and one parameterized constructor. Also define two method attributes for calculating the monthly and annual payment. Input the interest rate, the payment period (in years), and the loan amount. Create an object of the loan class using parameterized constructor in the main method. Use the "this" variable, to avoid any naming conflicts. Then obtain the monthly payment and the total payment using the instance methods in the Loan class. The following formula can assistant to calculate the monthlyPayment and totalPayment:

$$monthlyPayment = \frac{loanAmount \times monthlyInterestRate}{1 - \frac{1}{(1 + monthlyInterestRate)^{numberOfYears \times 12}}}$$

$$totalPayment = monthlyPayment \times numberOfYears \times 12$$

Please note that the annual interest rate is for the whole year and for monthly you will have to divide it by 12.

The running example is provided as follows:

```
Enter annual interest rate, for example, 8.25:2.5
Enter number of years as an integer:5
Enter loan amount, for example, 120000. 95:1000
The loan was created.
The monthly payment is 17.75
The total payment is 1064.84
```

Task 2: Body Mass Index (BMI)

Body Mass Index (BMI) is a measure of health based on height and weight. It can be calculated by taking your weight in kilogram and dividing it by the square of your height in meters. The interpretation of BMI for people 20 years or older is as follows:

BMI	Interpretation
BMI < 18.5	Underweight
18.5 ≤ BMI < 25.0	Normal
25.0 ≤ BMI < 30.0	Overweight
30.0 ≤ BMI	Obese

You need to implement one class called, BIM with all the properties and methods. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note that one pound is 0.45359237 kilograms and one inch is 0.0254 meters. Please note, if your BMI is calculated to 20.95 for the following example is also correct.

The example execution is provided as follows:

```
Enter weight in pounds:146
Enter height in inches:70
The BMI is 20.95
Normal
```

Task 3: Create a doc folder in your github repository. In the folder create a document named "Project_Description". In the document write the following:

I. Project Description

Decide and state which game your team will be working on.

II. Tool Name

Discuss and come up with a catchy name for your application

III. MoSCoW Matrix

Brainstorm on what features are critical for your project. Also discuss the properties that you good to have and would include if time permits. Submit the MoSCow matrix https://en.wikipedia.org/wiki/MoSCoW_method for your project. It should include

Must have

Should have

Could have

Wouldn't have

Submission Instructions

You need to submit **Task 1** and **Task 2** of the lab assignments at https://repl.it/. The assignments will be released in repl.it before every lab and you can submit your code for each of the assignment exercise. **Task 3 has to be uploaded to your respective git hub doc repository.** Create a folder in your doc repository by the name Week 2 and submit your Task 3 in the folder.

Deadline: Saturday 22nd Jan 2022 11:59 pm