University of Newcastle
School of Electrical Engineering and Computing
SENG1110/6110 Programming Assignment 1 – Semester 2, 2018
Due: By electronic submission (Blackboard) by 11:59pm on **Friday Sep 21**.

GOODMATES LOAN INSTITUTION

Introduction

The objective of this assignment is the implementation of an object oriented program using Java to receive information from a potential client and calculate and display details of a possible loan from the GoodMates Loan Institution. The assignment 2 will be an extension of this assignment.

Before you start

Carefully read the specification below. Make sure that you have all the information necessary to start the program. Do not assume what is necessary. There is a discussion board forum: assignment 1. Post your questions there and check it regularly. Start the assignment as soon as possible (after your computer lab this week). You will find the document "**HelpToStartAssign1**" and some Java files in Blackboard, which gives you some initial steps to start the assignment. I will answer questions and clarify any issue in the next week lecture (week 6).

Specification

The program will start asking for the following inputs:

- User **name**
- User **age**
- User **income** per year
- **amount** of money the user would like to borrow
- **number of months** the user would like to pay
- **type of account** (see below)

If the user inputs a negative number, the program should show a message and ask the input again.

There are two types of accounts the user can choose:

1) **No fees.** If the user chooses this option, the interest rate will be
   a. 6.5% if **n** < 50
   b. 7.5% if 50 <= **n** < =100
   c. 8.5% if **n** > 100
   where **n** is number of months

2) **Fees.** The interest rate will be **6%**, but the user needs to pay an additional fee of **$10** every month.

Next, the program will output the monthly payment, which will be calculated using the formula:

$$p = \frac{d * r * a^n}{12 * (a^n - 1)} \qquad (1)$$

where d = amount to borrow, a = 1+r/12, r = interest rate and n = number of months

Then, the program will show the amortization table. Each row of the table will show:

- **month**
- **initial balance**
- **payment**
- **interest paid**
- **principal paid**
- **final balance**

The **payment** is calculated by the formula (1).
The **interest paid** will be (**initial balance**)***r**/12.

The **principal paid** will be (**payment - interest paid**)
The **final balance** will be (**initial balance – payment + interest paid.)**

The program will also inform

- Total payment
- Total interest paid

For example:
Suppose the following inputs:

```
Name = Joao da Silva
Age = 31
Income per year = 60000
Amount to borrow = 10000
Months to pay = 10
Account type = no fees
```

The output would be:

```
Joao Silva, age 31, income 60000
Amount to borrow = 10000 to pay in 10 months in accountType "no fees".
The interest rate will be 6.5%.
The month payment will be p = $1030.03.
The total payment is 10300.34
Total interest paid is 300.34.
The amortization table is:

Month         InitBalance          MonthPayment  InterestPaid  PrincipalPaid FinalBalance
1             $10000.00            $1030.03      $54.17        $975.87       $9024.13
2              $9024.13            $1030.03      $48.88        $981.15       $8042.98
3              $8042.98            $1030.03      $43.57        $986.47       $7056.51
4              $7056.51            $1030.03      $38.22        $991.81       $6064.70
5              $6064.70            $1030.03      $32.85        $997.18       $5067.52
6              $5067.52            $1030.03      $27.45        $1002.58      $4064.94
7              $4064.94            $1030.03      $22.02        $1008.01      $3056.92
8              $3056.92            $1030.03      $16.56        $1013.47      $2043.45
9              $2043.45            $1030.03      $11.07        $1018.96      $1024.48
10             $1024.48            $1030.03       $5.55        $1024.48         $0.00
```

In the end, the program will ask the user if he/she would like to start the program again.

Program Requirements

There must be three classes: Client, Account and LoanCalculator.

- **The Account class** (the file needs to be **Account.java**)

  It will hold the required instance data for an account and it will have suitable methods to access and modify the data for an account.

  The instance variables will be
  - `interestRate` – double
  - `numberOfMonths` – int
  - `amount` – double
  - `accountType` – String

  You need to implement at least one constructor, which will initialize the instance variables with values from parameters. The class needs to have methods to change and access all instance variables. It will also have the following methods (at least):

  - `setInterestRate`, which will set the value in the interestRate which will depend on the type of the account (as explained above).
  - `calculateMontlyPayment`, which will calculate the formula (1)
  - `setAmortizationTable` - it will output a String with the table information.

- **The Client Class** (the file needs to be **Client.java**)

  It will hold the required instance data for a client and it will have suitable methods to access and modify the data for a client.

  The instance variables will be:
  - `name` – String
  - `age` – int
  - `income` – double
  - `loan` – Account

  You need to implement at least one constructor that will have the parameters name, age and income. The class needs to have methods to change and access all instance variables.

- **The LoanCalculator Class** (the file needs to be **LoanCalculator.java**)

  It will receive inputs and show outputs. It will have a Client variable. This is the only class that should have a **main method**. The class LoanCalculator will also be the only one that will receive inputs and show outputs.

  You can use TIO or GUI (it is your choice). On both cases, you should **use only the classes have seen in lectures**.

All the instance variables of your classes need to be **private** (this means that you are applying the principles of ***encapsulation***).

Marks will be awarded for: layout, both visual (variable names, indentation) and structural (scope of variables, use of methods); documentation (comments); and ability of the submission to perform as specified. A more detailed marking schema will be available in Blackboard.

You will find a document called **HelpToStartAssign1** and a template of all classes in Blackboard. You do not need to use these files neither follow the suggestions in the HelpToStartAssign1 document. It is optional.

What to submit.

You should submit the Java files (Client.java, Account.java and LoanCalculator.java) and the assignment cover sheet (available in Blackboard) electronically under the **Assignment 1** link in Blackboard. No .class files should be included in your submission, only .java files.

Extra Work for SENG6110 students

You must to implement the LoanCalculator class using GUI. You need to provide a UML class diagram of your program. You will find Powerpoint slides with an introduction of UML available in Blackboard in the area called 'SENG6110 extra material'. Ask Regina if you need any help.

Late Penalty and adverse circumstances

Note that your mark will be reduced by 10% for each day (or part day) that the assignment is late. This applies equally to week and weekend days. You are entitled to apply for special consideration because adverse circumstances have had an impact on your performance in an assessment item. This includes applying for an extension of time to complete an assessment item. See https://www.newcastle.edu.au/current-students/learning/assessments-and-exams/adverse-circumstances for more details.

In the Blackboard you will find a new forum in the discussion board: "assignment1". Any question about the assignment 1 you can post there. Check this forum regularly.

Prof Regina Berretta
Aug-2018