

ENGR 101 - Introduction to Programming

Mini Project 1

In this mini project, you are going to develop a basic Interest Calculator that reports monthly and total payments based on input annual interest rate. The goal of this project is to practice functions, for loops, operators, and getting user input through keyboard. The details about the calculator you are going to develop are provided below. (Note: example console outputs from the calculator are given as screenshots)

Compound Interest Calculator:

This calculator will be used by a single user. The calculator first prompts for and inputs information necessary for the final report such as loan amount, interest rate, and duration (see below for the complete list). Then a final report is being printed listing monthly and total payments over various durations as discussed below.

User Interface:

The app should first welcome the user as given below. The user will be asked to enter his/her name.

```
.....  
.*.*.*.*Welcome to the Interest Calculator*.*.*.*.*  
.....  
  
Please enter your name:Ali
```

The app will then prompt for and get the following inputs from the keyboard in the following order:

- annual interest rate (as a float percentage, e.g., 1.5 or 50 or 150 or ...)
- loan_amount (in \$s, e.g., 1000.50)
- max year (integer e.g., 0 or 1 or 2 or ...)
- max month (integer e.g., 0 or 6 or 18 or ...)
- iteration interval (integer in months)

You may assume that the user will always input a float or an integer when required by the program.

```
.....  
.*.*.*.*Welcome to the Interest Calculator*.*.*.*.*  
.....  
  
Please enter your name:Ali  
Loan amount:100  
Interest rate (per year):12  
-> TIME LENGTH  
  Loan term in years:1  
  Loan term in months:6  
  Iteration in months:3  
.....
```

Once user enters all necessary inputs, the final report should look like the following:

```
Report for Ali:
-----
-> year:0, month:3
Total payment:
103.0$
Monthly payment:
34.3$
-----
-> year:0, month:6
Total payment:
106.0$
Monthly payment:
17.6$
-----
-> year:0, month:9
Total payment:
109.0$
Monthly payment:
12.1$
-----
-> year:1, month:0
Total payment:
112.0$
Monthly payment:
9.3$
-----
-> year:1, month:3
Total payment:
115.0$
Monthly payment:
7.6$
-----
-> year:1, month:6
Total payment:
118.0$
Monthly payment:
6.5$
-----
```

Note that the report contains several entries. Each entry shows the total and monthly payment based on a payment plan over a different duration. The below entry, as an example, displays a plan for a period of three months.

```
-> year:0, month:3
Total payment:
103.0$
Monthly payment:
34.3$
```

Given the loan amount and interest rate, we use the below equation to calculate the associated total interest amount. Note that payments can easily be calculated with the interest amount.

$$\text{Total Interest} = (\text{loan_amount}/100) * (\text{int_rate}/12) * \text{month}$$

Implementation Details:

You will need to implement the following four functions to display the report. The naming and declaration of your functions should match with those given below:

1. **print_duration(month):** Given an integer for a duration in months, this function prints the duration in year and month format as follows:

Example:

```
print_duration(15)
>>> year:1, month:3
```

2. **print_money(money):** Given a float for a money amount, this function prints the amount with a single decimal digit, removing lesser significant digits. Finish with the dollar symbol (\$).

Example:

```
print_money(17.66666)
>>> 17.6$
```

3. **print_entry(loan_amount, int_rate, month):** This function prints a single entry. You should call Functions 1 and 2 within this function to print duration and money amounts respectively in the desired format.
4. **print_full_report(loan_amount, int_rate, max_year, max_mouth, iteration):** This function prints the whole report while making use of Function 3. The parameter iteration (in months) is the difference between the durations of two consecutive entries. The report displays entries up to a duration specified by the parameters max_year and max_mouth.

Warnings:

- **Do not** talk to your classmates on project topics when you are implementing your projects (This is serious). **Do not** show or email your code to others (This is even more serious). If you need help, talk to your TAs or the instructor, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve “**should**”, “**should not**”, “**do not**”, and other underlined/bold font statements.
- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.
- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code. You may be interviewed about any part of your code.
- You must use 4 function in this project. The use of skeleton function is given to you above. **If you don't follow this warning, you will lose 5 points for each function.**

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students (doing it individually is recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file. Name your code file with your and your partner's first and last names (see below for naming).
 - o If your team members are Deniz Barış and Ahmet Kemal Çalışkan, then name your code file as deniz_baris_ahmet_kemal_caliskan.py (Do **not** use any Turkish characters in file name).
 - o If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
 - o Those who **do not** follow the above naming conventions **will get 5 pts off** of their grade.
- Submit it online on LMS by **empty**

Late Submission Policy:

- -10%: Submissions between 17:01 – 18:00 on the due date
- -20%: Submissions between 18:01 – midnight (00:00) on the due date
- -30%: Submissions which are 24 hour late.
- -50%: Submissions which are 48 hours late.
- Submission more than 48 hours late will not be accepted.

Grading Criteria? :

Code Organization			Functionality				
Meaningful variable names (3 pts)	Compact code with no unnecessary repetitions (4 pts)	Sufficient commenting (4 pts)	Welcome message, prompts, user inputs (10 pts)	Print the report properly (25 pts)	Usage of 4 function properly (25 pts)	Correct calculation of total payment and monthly payment (20 pts)	Others (10 pts)

Have further questions?:

Please contact your TAs if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**

Good Luck!