Object-Oriented Programming and Design - Monsoon 2023

Assignment 1 – Basic Programming Practices

Arani Bhattacharya

This assignment carries a total of 60 marks. There are two questions, carrying 35 and 25 marks respectively.

1. Banks are often requested by their clients to explain how the value of money has changed over time. Unfortunately, most current banks do not have the tools to properly advise their customers. Assume that you are requested by an Indian bank to create a software that can estimate the value of money invested at any point of time beyond 1960. The value of money is given by the interest rate received every year on an investment minus the inflation rate. The dataset is given in the form of a comma-separated file. Your program needs to (i) read the interest rate and the inflation rate from the file, (ii) take input of the amount of investment and the year in which it was invested from the user, (iii) compute and print how much is the money actually received, (iv) compute and print how much value the money obtained after adding interest had in the year it was invested. Note that the money should be rounded to the nearest paisa in each case, and the number of unnecessary multiplications and divisions should be minimized in each case.

2. Now consider that a similar job was given by a bank of Spain. Spain replaced its currency Peseta by the Euro in 1999, and all investments automatically converted to this new currency in the beginning of 2002 at the rate of 1 euro = 166 Peseta. Solve the same problem as above, but with the assumption that the amounts invested are mentioned in the currency existing at the time the money was invested. Also, check if the value of money grew faster during the dictatorial era than the republican era. You should neither copy the source-code, nor have additional questions asked to the user about the country of the bank.

What and How To Submit

- The C++ program sources.
- `Makefile` to compile the source and generate the running binary for the shell. The Makefile should generate two versions of the binary – one for debugging and another for optimized execution.
- A readme text file, explaining the commands needed to build the file, and the format of the input files. If code is copied from anywhere else (not

that copying from any other student is plagiarism, but using textbook or open-source code is allowed), that should be mentioned here.

- At least 4 significant commits on a **private** github repository, with proper descriptions of the commits. You may have as many commits as you wish.
- At least 2 branches on github, one of them called main.
- Multiple functions dividing the program into logical units.
- Utilization of classes is optional. Utilization of templates and template libraries are not allowed.
- Checks on input for validity is compulsory.
- Make the assigned TA the admin of the github repository, **and** submit the same code in zipped form on Google Classroom by the due date.

Grading Rubric

For Q1:

1. Successful compilation using Makefile – 5 marks.
2. Ability to parse the file correctly – 5 marks.
3. Ability to take input from user and perform proper check – 5 marks
4. Proper computation (note that unnecessary computations will be penalized) – 5 marks
5. Proper printing of output, including rounding (needs to explain clearly what the value being printed is) – 5 marks
6. Proper utilization of functions – 5 marks
7. Proper utilization of git and github – 5 marks (Note that using a public github repository might lead to you losing all the marks in this assignment.)

For Q2:

1. Proper conversion of currencies in different corner cases – 5 marks
2. Proper extra output – 5 marks
3. Proper commits of code in branches – 10 marks
4. Proper utilization of functions, including proper names – 5 marks

In addition, there will be -5 in case of a missing readme file.

Late Submission Policy

- -0.25 per hour for the first 96 hours.
- Submissions beyond 4 days of delay would only be accepted with official leaves of absence.