Vietnam National University – HCMC
Hochiminh University of Technology
Faculty of Computer Science and Engineering

Programming Fundamentals

Assignment 01: HCS - A starting point

1 Introduction

With the development of technologies, we now have various ways to access health-care systems. One uses a smart watch can check their heart rate anytime, including the history of his movements. Some advanced IoT devices can even track your blood sugar level and send those information to provide a fast diagnosis. The health-care systems will soon become a part of our life in this digital world. In this assignment, the student has to build a very simple program that can simulate some features of a health-care system.

The student should perform a deep analysis of problem requirements before start implementing. In addition, the student is encourage to follow the problem solving process in order to do well in the assignment. For the first assignment, we focus on the basic user interface (UI) of the program. The student will apply their knowledge on I/O (input/output) programming, as well as using the if/else statements.

2 Requirements

For this assignment, a configuration is provided in form of a JSON file named "conf.json". Student don't have to worry about the format of this configuration file, however, the student should write the code to check what they are reading. The content of this configuration file is something like this:

```
{
  "name": "PF182-A01",
  "version": "1.0.0",
  "author": "Duc Dung Nguyen",
  "email": "nddung (at) hcmut.edu.vn",

  "WelcomeText": {
    "line1": "******************************************************************************",
    "line2": "*                      Welcome to CSE-HealthCare System                      *",
    "line3": "*     This is a simple application designed for the first assignment of PF   *",
    "line4": "* course (CO1011, Semester 182). The student must demonstrate the ability to *",
    "line5": "* write a program for a given problem. The student need to analyze the       *",
    "line6": "* requirements of the problem before implementing the application.           *",
    "line7": "******************************************************************************",
    "line8": "Email: nddung@hcmut.edu.vn",
    "line9": "(c) 2019 Duc Dung Nguyen All Rights Reserved."
  },
  "Menu": {
    "opt1": "Introduction",
    "opt2": "Login",
    "opt3": "Registration",
    "opt4": "Help",
    "opt5": "Exit"
  },
  "IntroTime": 3,
}
```

The content and indent of each line will follow the form of the above sample.

When the program is loading, the welcome message must be shown. The text is indicated in "WelcomeText" field. Then, our program must display a menu. The content of the menu is given in the "Menu" section. The menu must be displayed as follow:

```
1. Introduction
2. Log-in
3. Registration
4. Help
5. Exit
```

And the program will display a prompt to ask for an action from user:

```
Please select:
```

Please note that every content that you output on the terminal must be matched with the described output in order to pass the testcase.

When the user input their choice, the program must check if their input is correct or not. If the user input something wrong, which mean there was an error in the input stream, the program must output on the terminal a prompt:

```
Invalid input, please input an integer number.
```

If the user input a choice that is not recognized, the program must inform them:

```
Please select a number from 1 to 5.
```

After every wrong input, the program must show the menu and ask them to input again:

```
Please select:
```

If user select option 5, the program must output the following text and exit with code 0:

```
Exiting...
```

When user select a valid item in the menu, the program will execute the associated action of that item. For example, if user select item 1, the program will output:

```
You select menu item 1. Processing... Done!
```

Then the program must display the menu once again and wait for user input, until they decide to exit.

The following rules are applied:

- User must input an integer, floating point numbers are not accepted
- User can input any integer, including something like 0005
- Invalid inputs include 1a, 3., .6, a, g4, etc.

3 Implementation

An initial code package is provided with some source file and configuration files. The main.cpp file contain the main function of the application and should not be modified since it will be overwritten during grading process. There is another file named framecode.cpp that the student should not modify too. Two files core.cpp and core.h will be used for implementing features for this program.

In this first assignment, student can use provided functions, global variables. However, student can also write other functions if needed. Remember, file core.h is where you will write prototypes and core.cpp is the place to put your implementation.

4 Regulation

4.1 Evaluation

The program output will be compared with the expected output from the solution. A testcase is passed if everything in the program output is matched to the solution. The testcase will fail if the program runs too slow (timeout error). This issue should not occur in this assignment but the student must be aware.

4.2 Submission

Student should follows the submission instruction on the course site (e-learning site). One file src.zip contains core.cpp and core .h must be submitted during the given timeline. The OS system that builds and run code is Linux. Please do not use any special functions that does not exist on Linux, otherwise your code cannot be built. Student can build their code on Linux before submit to ensure that the code works.

The deadline for writing this assignment is 02-03-2019. The grading system will open at the deadline.

4.3 Rules

Student must perform this assignment themselves. Copying code is strictly prohibited and be considered as ethic code violation. In such case, the course result will be zero regardless the upcoming assignments. Please protect your code carefully. Student can help their friends by giving advise, not code.