ENGG 1110 Problem Solving By Programming 2017-2018 Second Term
Faculty of Engineering
The Chinese University of Hong Kong

Due date: 23 March 2018 (Fri) Project Phase 2 Full mark: 100 pts [30% of Project]

Course Registration Information System Phase 2 (curegis2)

Aim: 1. Improve your work done in Phase 1.
2. Practise writing and calling functions.
3. Practise using array and ordering elements.

Background:

This phase is a natural extension to our previous work. Functionally, we allow a student to register at most 9 ENGG courses in the shopping cart, and we perform GPA calculation from expected course grades. In terms of programming technique, we shall define and call C functions in this phase.

Requirements:

Design and implement the application in C.

- Firstly, create a new Code::Blocks Console application C project **`Phase2`**.
- Secondly, copy your own C program source file **`main.c`** from Phase1 to Phase2.
- Thirdly, copy sample functions and suggested function declarations from codeSubmit Project Phase 2 SUBMISSION. Assimilate the given code into your own work.
- Lastly, check your header comment block showing your name, SID, declaration on academic honesty, etc. Make sure you have also this header file inclusion line:

```
#include <string.h>                    // string functions header file
```

The program shall perform the following tasks to interact with the user and ***we shall assume user inputs are always integers, except the grade inputs that shall be strings of at most 2 characters***. Each user input is followed by an **Enter**.

a) Display a banner message.

b) Ask for SID in the form of 1166xxxxxx, i.e., a 10-digit integer with the prefix 1166. If the input integer does not conform, show an ERROR message and ask for SID again.

c) Initially, set up an empty course shopping cart, which can hold **at most 9 courses**.

d) Display the current shopping cart as well as a menu consisting of four choices:

1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out

Ask for an action code 1 – 4. If the input is out of range, show an ERROR message and repeat Step d).

e) Action 1. Add a course.
    1. If the shopping cart is full, show an ERROR message and go back to Step d).
    2. Ask for a course code in 1000 – 9999 inclusively. If the input is out of range, show an ERROR message and ask for the course code again.
    3. Check if the shopping cart has the course code to be added. If it is there already, show an ERROR message and go back to Step d).
    4. Keep the newly added course in the shopping cart, **in ascending order of the course code**.
    5. Go back to Step d).

f) Action 2. Drop a course.
    1. If the shopping cart is empty, show an ERROR message and go back to Step d).
    2. Ask for a course code in 1000 – 9999 inclusively. If the input is out of range, show an ERROR message and ask for the course code again.
    3. Check if the shopping cart has the course code to be dropped. If it is not there, show an ERROR message and go back to Step d).
    4. Remove the dropped course code from the shopping cart. Re-pack the shopping cart such that there shall be no empty slot in between and **the updated sequence of course code shall be in ascending order**.
    5. Go back to Step d).

g) Action 3. Clear shopping cart.
    1. Remove everything from the shopping cart.
    2. Go back to Step d).

h) Action 4. Check out.
    1. Check if the shopping cart is empty. If so, show an ERROR message and go back to Step d).
    2. Display SID and the final shopping cart with **course code in ascending order**.
    3. Ask the student to input expected grade (**[A/A-/B+/B/B-/C+/C/C-/D+/D], i.e., an 2-char case-insensitive string**) for each course in the shopping cart sequentially.
    4. **Assume all courses carry 3 credit units**, calculate and display expected GPA (with 2

decimal places).

5. Terminate the program and finish.

i) Sample and testing environment. A sample application for Windows is available on Blackboard. You shall download and try the sample rigorously to understand the requirements as well as the format of the menu, shopping cart, input prompts and ERROR messages.

Sample Run: (text in RED color indicates user input)

```
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
Course Registration Information System Phase 2
\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
Input your SID [in form of 1166xxxxxx]: 115501234
ERROR: Your SID is not conforming!
Input your SID [in form of 1166xxxxxx]: 1166123987
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Your shopping cart is empty!
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
1
==> 1. Add a course
Add course code [1000-9999]:
888888
ERROR: Invalid course code!
Add course code [1000-9999]:
1110
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Here is your shopping cart:
Course 1): ENGG1110
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
2
==> 2. Drop a course
Drop course code [1000-9999]:
2222
ERROR: Course code ENGG2222 not found in your shopping cart!
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Here is your shopping cart:
Course 1): ENGG1110
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
-1
ERROR: Wrong action -1
```

```
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
Here is your shopping cart:
Course 1): ENGG1110
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
1
==> 1. Add a course
Add course code [1000-9999]:
2222
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
Here is your shopping cart:
Course 1): ENGG1110
Course 2): ENGG2222
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
1
==> 1. Add a course
Add course code [1000-9999]:
3333
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
Here is your shopping cart:
Course 1): ENGG1110
Course 2): ENGG2222
Course 3): ENGG3333
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
1
==> 1. Add a course
Add course code [1000-9999]:
1000
================================================================================
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
Here is your shopping cart:
Course 1): ENGG1000
Course 2): ENGG1110
Course 3): ENGG2222
Course 4): ENGG3333
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
2
==> 2. Drop a course
Drop course code [1000-9999]:
2222
================================================================================
```

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
Here is your shopping cart:
Course 1): ENGG1000
Course 2): ENGG1110
Course 3): ENGG3333
Operation menu:
1. Add a course
2. Drop a course
3. Clear shopping cart
4. Check out
Action [1-4]:
4
==> 4. Check out
Dear student 1166123987
Here is your shopping cart:
Course 1): ENGG1000
Course 2): ENGG1110
Course 3): ENGG3333
Assume all courses carry 3 credit units, enter your expected grade for:
ENGG1000 [A/A-/B+/B/B-/C+/C/C-/D+/D]: B-
ENGG1110 [A/A-/B+/B/B-/C+/C/C-/D+/D]: B
ENGG3333 [A/A-/B+/B/B-/C+/C/C-/D+/D]: A+
ERROR: Invalid grade input!
ENGG3333 [A/A-/B+/B/B-/C+/C/C-/D+/D]: D+
Expected grade point for ENGG1000 = 2.70
Expected grade point for ENGG1110 = 3.00
Expected grade point for ENGG3333 = 1.30
Your expected GPA is 2.33
```

Guidance and Submission:

1. Check-out material made available on Blackboard and/or codeSubmit for you to understand the requirements.

2. Edit, Build (Compile), Run and Debug your program. If you do something wrong, don't panic. Double-click on the first error message. Check it, correct it and retry. Remember that a single mistake may trigger dozens of error messages. Always begin tackling the first error and conquer one at a time. Be reminded that the error message itself as well as the indicated line number may not be accurate.

3. Thoroughly Test Run your program with different input data sets such as extreme values and boundary values. You may try some of our test cases. However, you shall design your own test plan and test cases. Check and compare your program output carefully.

4. **Submit your code** in **`main.c`**, i.e., copy-and-paste, **to codeSubmit** to test your work and to get score by running our test cases.

5. Locate your Code::Blocks project folder **`Phase2\`** and
   Upload and Submit two files **`Phase2.cbp`** and **`main.c`** via Assignment Collection Box on **`https://blackboard.cuhk.edu.hk`**. We will examine your Code::Blocks project to assess you work on project management, coding style and/or running more tests.

Marking Scheme and Notes:

1. The submitted program should be free of any typing mistakes, compilation errors and warnings. Please make sure your submitted work can be compiled successfully on Code::Blocks, our

official platform.

2. Your proper declaration statement, your name and SID, file naming, adequate comment, code indentation, coding style such as variable naming, etc. are under assessment in every programming assessments unless specified otherwise.

3. Output formatting will be taken into account, e.g., word spellings, spaces, number formats, etc.

4. Starting this phase, the organization and structure of your work will be considered. For example, you shall define and call some functions as specified.

5. ***Remember to do your submission on BOTH codeSubmit AND Blackboard*** by the due date. Late submission made within another 72-hour grace period will be accepted with a 10-point late penalty. No further late submission may be accepted.

6. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be counted. You may delete (i.e. take back) your attached file and re-submit. We ONLY take into account the last submission.

7. Markers will check your work vigorously.

8. **If your last submitted source code on codeSubmit is different from your last submitted Code::Blocks source file on Blackboard, your work on codeSubmit may be discarded and got ZERO score.**

9. To guide you through this phase, there are some relevant practical exercises on codeSubmit.

University Guideline for Plagiarism

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details may be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students are required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.