CPE 100 Introduction to Computer Programming
Sections C and D August 2020
Laboratory Exercise 0

Objective

This exercise, which is suggested rather than required, allows you to practice using the C compiler to translate the source file for a C program into an executable program.

For information on using Linux commands, see ***http://windu.cpe.kmutt.ac.th/cpe100/LinuxIntro.html***.

Pre-requisites

To do this exercise, you need a computer that has a GCC-compatible C compiler installed. You also need a text editor program which will create plain text files. If you are using an IDE like Dev-C++, the editor is part of the environment. On Ubuntu Linux, the default editor is called **gedit**.

For information about GCC-compatible environments on different operating systems, see the About Compilers link on the CPE100 home page.

Instructions

1. Start your computer. If you are using an IDE, start the IDE program. Otherwise, if you are using a command-line-based compiler, open a terminal window.
2. Create a new file called **grade.c**.
3. Type in the C code on the next page (adapted from, but not exactly the same as, code shown in Lecture 1). You must type the program **exactly** as shown. ***Do not use copy and paste!*** Make sure that you have the correct punctuation.
4. Compile your code into an executable program. If you are using a terminal and the command line, the command to do this is:

   gcc -o grade grade.c

   (You may see error messages if you did not copy the code exactly. In this case, fix the errors, then try again.)
5. Run your program and test it. To do this in a terminal, type:

   ./grade

6. When you are sure that your program is working correctly, copy the file **grade.c** to a new file called **newgrade.c**.
7. Edit **newgrade.c**. Increase the number of quizzes the program expects from 5 to 9. (Don't forget to change the statement that calculates the average). Also change the grade that indicates a failure from 60 to 50. That is, the student passes if her grade is 50 or higher, otherwise she fails. Also - ***add a line to the comment at the top that includes your name and your student ID (s***ince you are now the author of this program).
8. Compile **newgrade.c** to create an executable program. Run the executable **newgrade.** Test that your changes work correctly. One way to do this is to enter values that make the correct answers easy to predict.

Upload the file **newgrade.c** by going to the course web page, ***http://windu.cpe.kmutt.ac.th/cpe100,*** and clicking on the link labeled "*Upload work to server*". Follow the directions on the web form that will appear.

```
/* Calculate a student's grade.
 * Ask the user for quizzes, midterm and final exam scores,
 * then print the results. Quizzes count 25%, midterm 25%,
 * final exam 50%. Final grade is a number, not a letter.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
   char stringInput[128];   /* get input from the terminal */
   int totalScores = 0;     /* sum of quiz scores */
   int quizScore = 0;       /* one quiz score */
   int midScore = 0;        /* midterm exam score */
   int finalScore = 0;      /* final exam score */
   int averageScore = 0;    /* average of quizzes */
   int grade = 0;           /* numeric course grade */
   int quiz;                /* counter for quizzes */

   /* Calculate average of quizzes */
   for (quiz = 1; quiz <= 5; quiz++)
      {
      printf("Enter score for quiz number %d: ", quiz);
      fgets(stringInput, sizeof(stringInput), stdin);
      sscanf(stringInput,"%d",&quizScore);
      totalScores = totalScores + quizScore;
      }
   averageScore = totalScores/5;

   /* Get the midterm exam score */
   printf("Enter midterm exam score: ");
   fgets(stringInput,sizeof(stringInput), stdin);
   sscanf(stringInput,"%d",&midScore);


   /* Get the final exam score */
   printf("Enter final exam score: ");
   fgets(stringInput,sizeof(stringInput), stdin);
   sscanf(stringInput,"%d",&finalScore);

   /* Calculate overall grade */
   grade = averageScore * 0.25 + midScore * 0.25 + finalScore * 0.5;
   printf("Your grade is %d\n", grade);
   if (grade < 60)
      printf("Sorry, you've failed the course.\n");
   else
      printf("Congratulations! You passed the course.\n");
   exit(0);
}
```