

Computer Science 1081 – Assignment #07

Program #1:

Write a program that uses a function *displayMenu* to display the following menu:

Math Program:

1. Add two numbers
2. Subtract two numbers
3. Multiply two numbers
4. Divide two numbers
5. Quit

When the user selects one of the options, the program will prompt them to input two numbers. It will then pass these numbers to a function, which will do the calculation and return the desired result.

- There should be one function for each of the four non-quit menu options. The program will then output the correct answer.
- The function names should be named in a way that corresponds to their action (add, subtract, multiply, and divide).
- All of these functions should accept two floating point numbers, and return a floating point number.
- Display all results to 2 significant digits.
- All input and output with the user should be done in the main function.
 1. The user is expected to input both numbers at the same time
- The program should loop until the user selects option 5 (Quit), and should be notified if they do not pick an option between 1 and 5.

Sample Output:

Math Program:

1. Add two numbers
2. Subtract two numbers
3. Multiply two numbers
4. Divide two numbers
5. Quit

Choice: 0

You must choose between 1 and 5!

Math Program:

1. Add two numbers
2. Subtract two numbers
3. Multiply two numbers
4. Divide two numbers
5. Quit

Choice: 2

Please enter two numbers: 4.5815 5.8481

The numbers subtracted are -1.27

Math Program:

1. Add two numbers
2. Subtract two numbers
3. Multiply two numbers
4. Divide two numbers
5. Quit

Choice: 3

Please enter two numbers: 8 2

The numbers multiplied are 16.00

Math Program:

1. Add two numbers
2. Subtract two numbers
3. Multiply two numbers
4. Divide two numbers
5. Quit

Choice: 5

Thanks for using the program!

Press any key to continue . . .

Program #2

Create a function *repchar(char ch, int n)* that has two default arguments.

- You can define the default arguments in either the function prototype or the function definition (if you do not use a prototype for your function)
- The function will display the character in *ch*, *n* number of times. Use '*' as the default argument for *ch* and 45 as the default argument for *n*.
- Call *repchar* three times in the main function as follows:
 1. With no arguments
 2. '=' as the single argument
 3. '+' and 30 as the arguments
- Do not print anything else from your function.

Sample Output:

```
*****
=====
+++++
Press any key to continue . . .
```

Program #3

Create three definitions for an overloaded function *repchar()*.

- Call each of the *repchar* definitions as follows:
 1. `repchar();`
 2. `repchar('=');`
 3. `repchar('+', 30);`
- This program will be different from Program 2, as you will not use any default arguments, and will instead overload the function *repchar* three times.

Sample Output:

```
*****
=====
+++++
Press any key to continue . . .
```

Program #4

Write a program that calculates the average of a group of test scores, where the lowest score in the group is dropped. It should use the following functions:

- Void `getScore()` should ask the user for a test score, store it in a reference parameter variable, and validate it. This function should be called by main once for each of the five scores to be entered.
- Void `calcAverage()` should calculate and display the average of the four highest scores. This function should be called just once by main and should be passed the five scores.
- Int `findLowest()` should find and return the lowest of the five scores passed to it. It should be called by `calcAverage`, which uses the function to determine which of the five scores to drop.
 - The main function should define five variables, call *getScore* for each variable, call *calcAverage* passing all five variables, and exit
 - *calcAverage* should call *findLowest*
 - *findLowest* should return the value of the score to be dropped
 - You can then use this information to adjust the total score before averaging
 - Do not accept scores lower than 0 or higher than 100
 - Loop until the user complies
 - Display the average to 3 significant digits

Sample Outputs:

```
Enter a test score: 81
Enter a test score: 95
Enter a test score: 73
Enter a test score: 88
Enter a test score: 94
After dropping the lowest score, the test average is 89
Press any key to continue . . .
```

```
Enter a test score: -151
Enter a test score between 0 and 100: 51
Enter a test score: 151
Enter a test score between 0 and 100: 54
Enter a test score: 65
Enter a test score: 35
Enter a test score: 75
After dropping the lowest score, the test average is 61
Press any key to continue . . .
```