

# project[4]: Interactive Programs

Due Thursday, 10/12/2017, 12:00pm

## Project Goals

The goals of this project are to:

1. Get you familiar with switch statements
2. Get you familiar with writing interactive programs
3. Get you familiar with character input/output functions

### Important Notes:

1. **Formatting:** Make sure that you follow the precise recommendations for the output content and formatting: for example, do not change the text in the first problem to differ from what is in the assignment. Your assignment will be auto-graded and any change in formatting will result in a loss in the grade.
2. **Comments:** Header comments are required on all files and recommended for the rest of the program. Points will be deducted if no header comments are included.

Write an interactive program that implements a simple calculator. The program should allow the user to choose a binary arithmetic operation and enter two terms to which to apply the operation. The program should then compute the result and display it to the user.

At the beginning, the program should output the following information:

This program implements a calculator. Options:

- 1 - addition
- 2 - subtraction
- 3 - multiplication
- 4 - division
- 5 - toggle precision
- 6 - exit program

At every iteration, your program should ask the user for an option as follows:

Please enter your option:

Below is a description of what the program should do for each of the above options:

1. Perform addition. For this option, the program should ask the user to enter two terms to be added, as follows:

```
Enter first term: 1
Enter second term: 1
The sum is: 2.000000
```

2. Perform subtraction. For this operation, the program should ask the user to enter two terms to be subtracted, as follows:

```
Enter first term: 4
Enter second term: 2
The difference is: 2.000000
```

3. Perform multiplication. For this operation, the program should ask the user to enter two terms to be multiplied, as follows:

```
Enter first term: 2
Enter second term: 3
The product is: 6.000000
```

4. Perform division. For this operation, the program should ask the user to enter two terms to be divided, as follows:

```
Enter first term: 4
Enter second term: 2
The quotient is: 2.000000
```

If the second term entered by the user is 0, the program should print

```
Cannot divide by zero!
```

5. Toggle the precision used for computation. By default, the program should perform its calculations using single-precision floating point arithmetic. However, if the user chooses this option, the program should change its precision: either from float to double or vice versa.

When switching from single to double precision the program should print the following message:

```
Calculator now works with double precision.
```

When switching from double to single precision the program should print the following message:

```
Calculator now works with single precision.
```

In single precision mode, the program should print the results with 6 digits after the decimal point and in double precision, the answers should be printed with 15 digits after the decimal point.

6. Exit. For this option, the program should end.

**Note:** If the user enters an invalid command, the program should print the menu again and ask for a valid option (same as it is done in the beginning).

Save your program in a file called `calc.c`

**Challenge 1 (5 extra credit points):** Turn your calculator into a scientific calculator. Add options 7 and 8 as follows:

7. Compute  $\exp(x)$  . For this option, the program should prompt the user to enter a single number as follows:

```
Enter term: 4  
The result of  $\exp(x)$  is: 54.598150
```

8. Compute  $\log(x)$  . For this option, the program should prompt the user to enter a single number as follows:

```
Enter term: 4  
The result of  $\log(x)$  is: 1.386294
```

Note that this is the *natural* logarithm. If the user enters a negative number, your program should print

```
Cannot take the log of a negative number!
```

Save your program in a file called `calc_c.c`

## Grading Rubric

Grading will be done for each problem as follows:

<b>Correctly-named file</b>	5%
<b>Header comment</b>	2%
<b>Program compiles</b>	5%
<b>Correctly-reading data from terminal</b>	28%
<b>Correct result printed</b>	60%

## Submission details

Follow the new instruction procedures used in the previous lab. The submission script copies all files in the current directory to our directory. You may submit as many times as you like before the deadline, we only keep the last submission.

## Academic Honesty

Academic dishonesty is against university as well as the system community standards. Academic dishonesty includes, but is not limited to, the following:

Plagiarism: defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion.

Cheating: defined as (1) obtaining or providing unauthorized information during an examination through verbal, visual or unauthorized use of books, notes, text and other materials; (2) obtaining or providing information concerning all or part of an examination prior to that examination; (3) taking an examination for another student, or arranging for another person to take an exam in one's place; (4) altering or changing test answers after submittal for grading, grades after grades have been awarded, or other academic records once these are official.

Cheating, plagiarism or otherwise obtaining grades under false pretenses” constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include canceling a student’s enrollment without a grade, giving an F for the course, or for the assignment. For more details, see the University of Nevada, Reno General Catalog.