

Course: CISC 105: General Computer Science, Honors Section
Semester: Fall 2006
Professor: Chandra Kambhamettu
Due: 9/28/2006, 10pm. (submit the printout in the next class)
Total points: 100+10.

Programming Assignment 1 - Basic Math Instructor

1 Academic Dishonesty

Please be sure to re-read the syllabus on Academic Dishonesty. For all programming assignments, students must work INDEPENDENTLY.

2 Grading

80% of the grade depends on your program's correctness and 20% of the grade depends on your program's readability. Your program style should conform with the programs shown in the textbook. Use meaningful variable names and provide indentation to improve readability of your program.

3 Objectives

- Write an algorithm showing the list of steps to take for solving the given problem. In this process, you should identify the functions that are to be used in the program.
- Develop a C program based on the designed algorithm.
- Correct the program for any syntax errors.
- Test the program extensively, for any logic errors.

4 Assignment

4.1 Overview

Computers are playing an increasing role in education. Write a program that will help an elementary school student learn addition, subtraction, multiplication and division. Use a random number generator to generate two positive numbers between 1 and 20 in order to ask queries such as "How much is 6 times 7?", "How much is 10 added to 20?", "How much is 16 subtracted from 19?", "How much is 8 divided by 4?" etc.

The student then types the answer. Your program checks the student's answer. If it is correct, you can say "Very Good!!". If the answer is wrong, you can say "Try again: ". You can ask a maximum of 3 times before giving out the answer.

4.2 Program Design

The program design should make extensive use of functions. Here is an outline of what `main()` should do.

1. Invoke a function `show_instructions` which displays a welcome message and general instructions to the user.
2. Prompt the user for the input: a, s, m, d, q (a: addition, s: subtraction, m: multiplication, d: division, q: quit). Invoke the functions appropriately.
2. For each arithmetic operation, ask the user 4 different number combinations.
3. In order to generate the numbers, invoke a random-number generation function, `random` which will generate a number between 1 and 20 (inclusive).
4. In case of division, only use integers which are divisible. i.e, numbers such as 4 divided by 2, 6 divided by 3, 20 divided by 10 etc. DO NOT use 3 divided by 6, where the answer is 0.5, a float, or 19 divided by 7, etc. Hint: You can use `%` operator to identify the divisible numbers.

4.3 Technical Details

Use the following statements to generate the two random numbers between 1 and 20.

```
number1 = rand () % 20 ;  
number2 = rand () % 20 ;
```

You need to add `#include <stdlib.h>` at the top of the program, along with `#include <stdio.h>` in your program. This will have the definition of `rand()` function.

4.4 Sample Run

```
-----  
Welcome to the Basic Math Instructor!! This program allows you to assess your  
math skills in addition, subtraction, multiplication and division.  
-----
```

Type any of the following alphabet now.

```
a for addition  
s for subtraction  
m for multiplication  
d for division  
q for quit the program
```

Type-in now: a

You have chosen Addition!!

Tell me how much is $6+9$? 15

Very good!

Tell me how much is $7+10$? 16

Sorry, try again: 17

Very good!

Tell me how much is $15+19$? 23

Sorry, try again: 22

Sorry, try again: 21

Sorry, try again: 26

Oops! You missed this one. The correct answer is 34.

Tell me how much is $20+10$? 30

Very good!

----- Now, I am sending you back to the main menu -----

Type any of the following alphabet now.

a for addition

s for subtraction

m for multiplication

d for division

q for quit the program

Type-in now: q

5 What to Hand In

Run your program for each operation. Include 2 correct and 2 incorrect student responses for each operation, and make a script file for grading. For those doing extra credit, run the program (with 2 correct and 2 incorrect student responses for each operation) on interactive mode, and atleast 2 different I/O filenames.

6 Extra Credit (10 points)

Ask the user if files are to be used for input/output. If yes, request the user for a filename to read and a filename to write out. Use file I/O to read and write using the specified files instead of interactive I/O.

Create files such as add_input, sub_input, etc. Instead of using random number generator, create the numbers in these input files. Write out the output to add_out, etc., while interactively generating the output on the screen as well.