

CMIS 102 Hands-On Lab

Week 4

Overview

This hands-on lab allows you to follow and experiment with the critical steps of developing a program including the program description, analysis, test plan, design (using pseudocode), and implementation with C code. The example provided uses sequential and selection statements.

Program Description

This program will calculate the sum of 5 integers. The program will ask the user to input 5 integers. If the sum of the numbers is greater than 100, a message is printed stating the sum is over 100. The design step will include both pseudocode and C code.

Analysis

I will use sequential, and selection programming statements.

I will define six integer variables: value1, value2, value3, value4, value5 and sum. The value1, value2, value3, value4 and value5 variables will store the integers input by the user. The sum will store the sum of the 5 values.

The sum will be calculated by this formula:

$$\text{sum} = \text{value1} + \text{value2} + \text{value3} + \text{value4} + \text{value5}$$

For example, if the first values entered were 1,1,2,2 and 3 respectively:

$$\text{sum} = 1 + 1 + 2 + 2 + 3 = 9$$

The additional selection statement will be of this form:

```
If sum > 100 then
    print "Sum is over 100"
End If
```

Test Plan

To verify this program is working properly the input values could be used for testing:

| Test Case | Input | Expected Output |
|-----------|---|--------------------------------|
| 1 | value1 = 1 value2 = 1 value3 = 1 value4 = = value5 = 2 | Sum is 5 |
| 2 | value1=100 value2=100 value3=100 value4=100 value5=200 | Sum is 600 Sum is over 100. |
| 3 | value1= -100 value2= -100 value3= -200 value4 = 0 value5= 200 | Sum is -200 |

Pseudocode

```
// This program will calculate the sum of 5 integers.

// Declare variables
Declare value1, value2, value3, value4, value5, sum as Integer

// Enter values
  Print "Enter an Integer for value1"
  Input value1
  Print "Enter an Integer for value2"
  Input value2
  Print "Enter an Integer for value3"
  Input value3
  Print "Enter an Integer for value4"
  Input value4
  Print "Enter an Integer for value5"
  Input value5

// Calculate sum
Set sum = value1 + value2 + value3 + value4 + value5

// Print results and messages
Print "Sum is " + sum
If (sum > 100) Then
  Print "Sum is over 100"
End if
```

C Code

The following is the C Code that will compile in execute in the online compilers.

```
// C code
// This program will calculate the sum of 5 integers.
// Developer: ENTER YOUR NAME
// Date: ENTER DATE YOU FINISH PROGRAM

#include <stdio.h>

int main ()
{
    /* variable definition: */
    int value1,value2,value3,value4,value5,sum;

    printf("Enter an Integer for value1\n");
    scanf("%d", &value1);
    printf("Enter an Integer for value2\n");
    scanf("%d", &value2);
    printf("Enter an Integer for value3\n");
    scanf("%d", &value3);
    printf("Enter an Integer for value4\n");
    scanf("%d", &value4);
    printf("Enter an Integer for value5\n");
    scanf("%d", &value5);

    sum = value1 + value2 + value3 + value4 + value5;

    printf("Sum is %d\n " , sum );
    if (sum >100)
        printf("Sum is over 100\n");
    return 0;
}
```

Setting up the code and the input parameters in ideone.com:

Note the input integer values are 100, 100, 100, 200 and 100, for this test case. You can change these values to any valid integer values to match your test cases.

Results from running within ideone

The screenshot shows a web browser window with the URL `http://ideone.com/4B0dXi`. The page content includes a code editor with the following C code:

```
1. // C code
2. // This program will calculate the sum of 5 integers.
3. // Developer: J.Marcus
4. // Date: March 26, 2016
5.
6. #include <stdio.h>
7.
8. int main ()
9. {
10.     /* variable definition: */
11.     int value1,value2,value3,value4,value5,sum;
12.     /* Initialize sum */
13.     sum = 0;
14.     printf("Enter an Integer for value1\n");
15.     scanf("%d", &value1);
16.     printf("Enter an Integer for value2\n");
17.     scanf("%d", &value2);
18.     printf("Enter an Integer for value3\n");
19.     scanf("%d", &value3);
20.     printf("Enter an Integer for value4\n");
21.     scanf("%d", &value4);
22.     printf("Enter an Integer for value5\n");
23.     scanf("%d", &value5);
24.
25.     sum = value1 + value2 + value3 + value4 + value5;
26.
27.     printf("Sum is %d\n ", sum );
28.     if (sum >100)
29.         printf("Sum is over 100\n");
30.     return 0;
31. }
32.
```

On the right side of the page, there is a Kony advertisement with the text "Kony - It's Mobile Time" and "5 Great Steps To Mobile Strategy. Read The Free E Book!". Below the advertisement, the page metadata is displayed: language: C, created: 1 minute ago, and visibility: public. There are also social sharing icons for Facebook, Twitter, Google+, and others.

```
29.     printf("sum is over 100\n");
30.     return 0;
31. }
32.
33.
```

Learn How to Code

Success #stdin #stdout 0s 2116KB comments (0)

CLAIM-FREE REWARDS:
Get money back every year you don't have a claim.
[LEARN MORE](#)

stdin [copy](#)

```
100
100
100
200
100
```

stdout [copy](#)

```
Enter an Integer for value1
Enter an Integer for value2
Enter an Integer for value3
Enter an Integer for value4
Enter an Integer for value5
Sum is 600
Sum is over 100
```

Feedback

Learning Exercises for you to complete

1. Enter the baseline program into a compiler and run it. Provide readable screen captures showing the code of the full program (including comments), the input data provided, and the results of the program running.
2. Change the C code to sum 10 integers as opposed to 5. Do not use arrays or loops for this; those features will be covered in the future. Support your experimentation with a screen capture showing the full listing of your program (including all comments), and executing the new code
3. Using the code you create in step 2, modify the code to print an additional statement if the sum of the value is negative. Support your experimentation with a screen capture of showing a full listing of the revised code (including all comments), and executing the new code.
4. Prepare a new test table with at least 3 distinct test cases listing input and expected output for the code you created after step 3. Be sure that your test data tests all possible logic paths in the program.
5. Create your own C code implementation of one of the following mathematical formulas:
 - a. $y = mx + b$; (line equation) The user will be prompted to input float point values for m, x and b and the program will calculate y. If the value of y is greater than 10, inform the user the value is greater than 10.
 - b. $a = \text{PI} * r * r$; (area of circle). The user will be prompted to input the radius r. Do not input PI, define PI as 3.14159 inside the program. If the value of a is greater than 10, inform the user the value is greater than 10.
 - c. $v = \frac{4}{3} \text{PI} * r * r * r$; (volume of sphere) The user will be prompted to input the radius r. Do not input PI, define PI as 3.14159 inside the program. If the value of v is greater than 10, inform the user the value is greater than 10.

Be sure you provide not only the C code but a test table with at least 3 distinct test cases listing inputs to the program and expected output of the mathematical formula. Be sure that your test data checks every possible logic path in your program.

Submission

Submit a neatly organized Word document that demonstrates you successfully executed this lab on your machine using an online compiler. You should provide a screen capture of the resulting output. Submit all C code you created in files.

Also, provide the answers and any associated screen captures of your successful completion of exercises 1, 2, 3, 4 and 5.

Submit your document no later than the due date listed in the syllabus or calendar.

| Grading guidelines Submission | Points |
|---|-----------|
| Successfully demonstrates execution of this lab with online compiler. Includes a program listing (including comments) and a screen capture showing the input data and the program executing. | 2 |
| Modifies the C code to sum 10 integers as opposed to 5. Supports your experimentation with a program listing (including comments) and a screen capture showing the input data and the program executing. | 2 |
| Using the code created in step 2, modifies the code to print an additional statement if the sum of the value is negative. Supports your experimentation with a full listing of the revised code (including comments) and a screen capture of executing the new code. | 2 |
| Provides a new test table with at least 3 distinct test cases listing input and expected output for the code you created after step 3. Test data checks every possible logic path through program | 1 |
| Creates your own unique C code implementation of one of the provided mathematical formulas. Provides a new test table with at least 3 distinct test cases listing input and expected output your mathematical formula. Supports your experimentation with a full listing of the program (including all comments) and a screen capture of executing the new code. | 2 |
| Document is well-organized, and contains minimal spelling and grammatical errors. | 1 |
| Total | 10 |