CMIS 102 Hands-On Lab

Week 7

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, and implementation with C code. The example provided uses sequential, repetition, selection statements and two user-defined function.

Program Description

This program will provide options for a user to calculate the square or cube of a positive Integer input by a user. The program will prompt the user to enter an Integer and then prompt the user if they want to calculate the square of the cube of the number. Based on the inputs of the user, the program will output the square of the cube of the positive integer. The program will then print the Integer and square or cube of the integer based on the user's original choice. The program will continue to prompt the user for Integers and their calculation choice until the user enters a negative integer. The square and cube calculations should be calculated using a function.

Analysis

I will use sequential, selection, and repetition programming statements and functions for the cube and square calculations.
I will define three Integer numbers: IntValue, MenuSelect, Results to store the Integer value input by the user, the Menu selection (1 for Square, 2 for Cube) of the user, and the results of the Square or Cube functions.
The Square function will take one Integer as input and return one Integer as the output. The calculation within the Square function is: Results = IntValue * IntValue
For example, if 10 was entered as the IntValue. Results = 10*10 = 100
The Cube function will take one Integer as input and return one Integer as the output. The calculation within the Cube function is: Results = IntValue * IntValue*IntValue
For example, if 10 was entered as the IntValue. Results = 10*10*10 = 1000

A repetition loop can be used to loop through iterations until a negative is entered:

```
while(intValue > 0) (
  …
End For
```

Test Plan

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

Test Case	Input	Expected Output
1	IntValue=10 MenuSelect=1	Square of 10 is 100
2	IntValue=10 MenuSelect=2	Cube of 10 is 1000
3	intValue=-1 MenuSelect=N/A	Program exits

Pseudocode

```
// This program will provide options for a user to calculate the square

// or cube of a positive Integer input by a user.
// Start Main program
 Main

// Declare variables
   Declare intValue, menuSelect,Results as Integer

// Set intValue to positive value to start loop
   Set intVal = 1;

   // Loop While input is a positive number

 While intValue > 0

      Print "Enter a positive Integer:”
      Input intValue

      // Only perform menu and function calls is integer is positive
     If intValue > 0 Then
     Print "Enter 1 to calculate Square, 2 to Calculate Cube "
     Input menuSelect

     If menuSelect == 1   Then
          // Call the Square Function
          Set Results = Square(intValue)
          Print intValue,Results
      Else If  menuSelect == 2 Then
         // Call the Cube function
         set Results = Cube(intValue)
         Print intValue,Results
      Else
         Print “Invalid menu item, only 1 or 2 is accepted”
         End If
      End If

    END While

// End of Main program
End Program
```

```
// Square Function
Function Square(value) as Integer
    Set Square = value*value
End Function

 // Cube Function
Function Cube(value) as Integer
    Set Cube = value*value*value
End Function
```

C Code

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

```
// C code

// This program will provide options for a user to calculate the square

// or cube of a positive Integer input by a user.

// Developer: Faculty CMIS102

// Date: Jan 31, XXXX

#include <stdio.h>

// Function prototypes
int Square(int value);
int Cube(int value);


int main ()

{

  /* variable definition: */

   int intValue, menuSelect,Results;

   intValue = 1;

   // While a positive number

 while (intValue > 0)

  {

     printf ("Enter a positive Integer\n: ");

     scanf("%d", &intValue);

    if (intValue > 0)

    {

     printf ("Enter 1 to calculate Square, 2 to Calculate Cube \n: ");

     scanf("%d", &menuSelect);
```

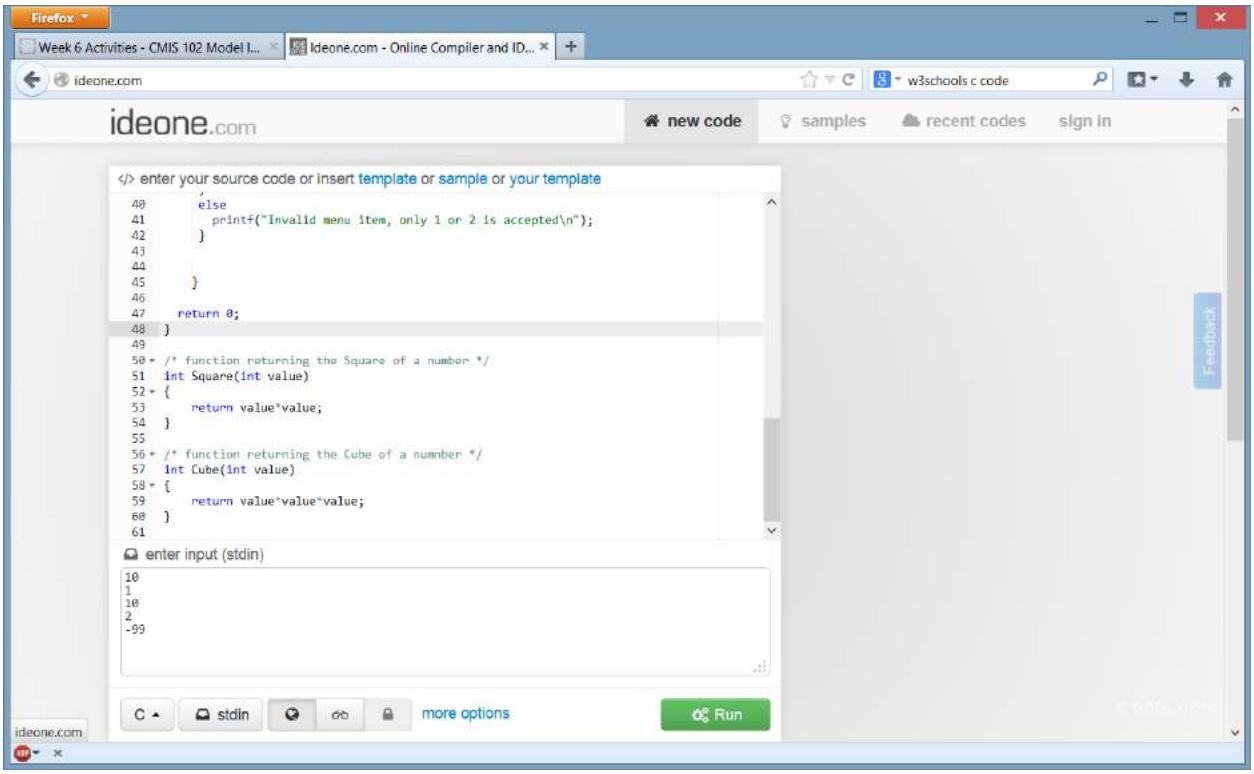
```
      if (menuSelect == 1)
      {
         // Call the Square Function
         Results = Square(intValue);
         printf("Square of %d is %d\n",intValue,Results);
      }
      else if (menuSelect == 2)
      {
         // Call the Cube function
         Results = Cube(intValue);
         printf("Cube of %d is %d\n",intValue,Results);
      }
      else
        printf("Invalid menu item, only 1 or 2 is accepted\n");
      }
     }
  return 0;
}
/* function returning the Square of a number */
int Square(int value)
{
    return value*value;
}
/* function returning the Cube of a number */
int Cube(int value)
{
    return value*value*value;
}
```

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

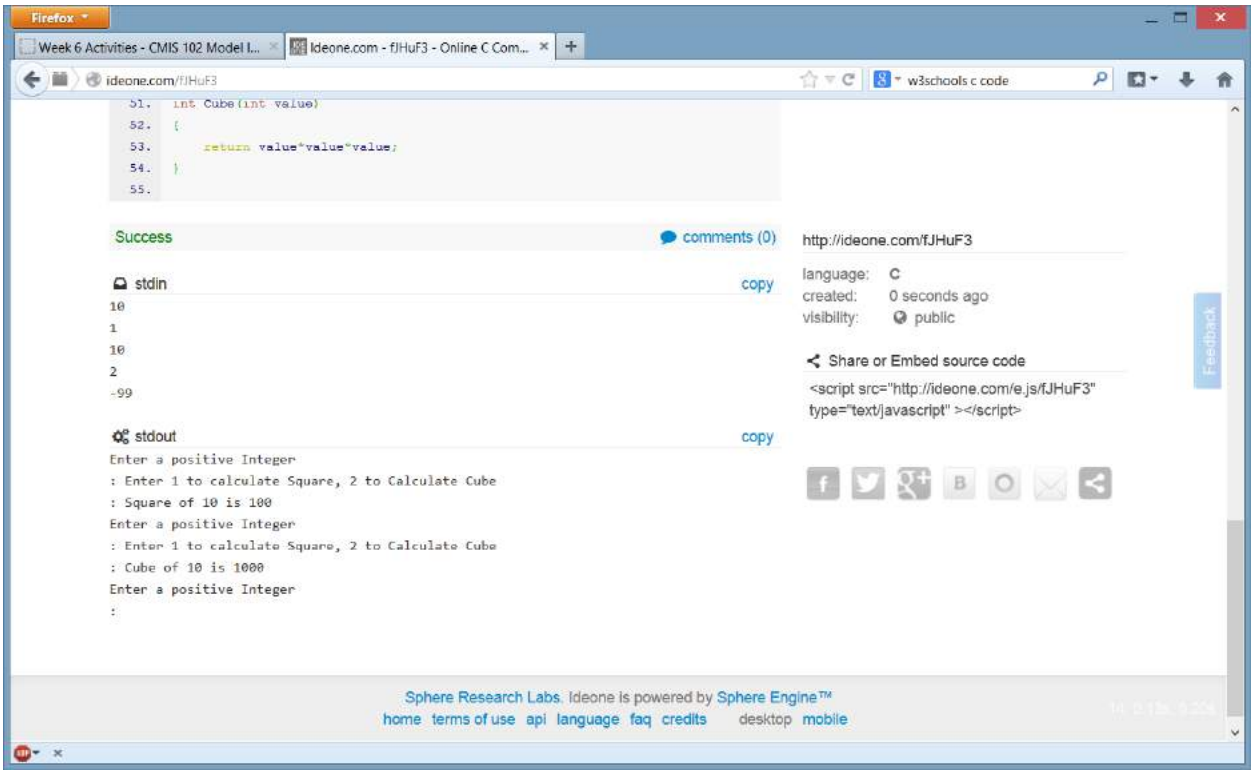
Note the Input values for this run were:
10

1
10
2
-99

You can change these values to any valid integer values to match your test cases.



Results from running the programming at ideone.com:



Learning Exercises for you to complete

1. Using the Square and Cube functions as models, Create an application that includes a function named “Shrink” that would take an Integer and return the Integer divided by 2? (Hint: your returned value should probably be a double type.) Support your experimentation with screen captures of executing the new code.

2. Prepare a new test table with at least 3 distinct test cases listing input and expected output for the code you created after step 1.

3. What would happen if you removed the following code from our design?

   If intValue > 0

   Support your argument with screen captures of executing the new code.

4. Modify the original code and add an additional function of your choice. The function should be unique and something you created for this assignment. Support your experimentation with

screen captures of executing the new code. Prepare a test table with at least 3 distinct test cases listing input and expected output for your unique function.

Grading guidelines

Submission	**Points**
Successfully demonstrates execution of this lab with online compiler. Includes a screen capture.	2
Modifies the code to create an application that includes a function named “Shrink” that would take an Integer and return the Integer divided by 2 Support your experimentation with screen captures 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 1.	1
Describes what would happen if you removed the “if intValue = 0“ line was removed. Support your argument with screen captures of executing the new code.	1
Modifies the original code and adds an additional unique function of your choice. Supports your experimentation with screen captures of executing the new code. Prepares a test table with at least 3 distinct test cases listing input and expected output for your unique function.	3
Document is well-organized, and contains minimal spelling and grammatical errors.	1
Total	**10**