Laboratory 05A

CS-102

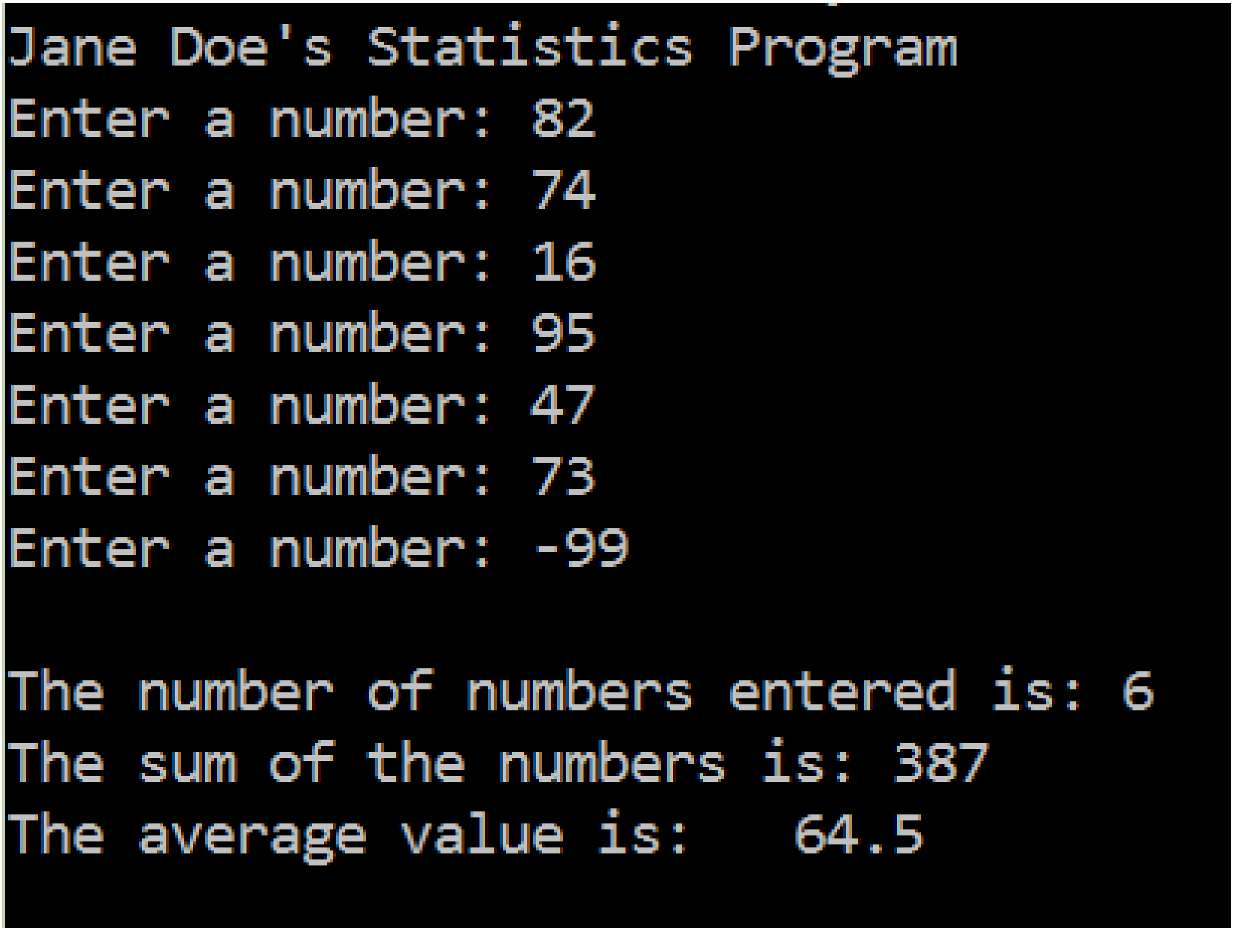
Spring 2022

Laboratory 5A

- Write a program with a loop that lets the user enter a series of integers until the user enters the number -99.
- When that occurs, the program should print out:
  - The number of integers that you entered.
  - The sum of all the integers that you entered.
  - The average of all the integers that you entered.
- Your program should also test to be sure that the number of integers that you typed in is > 0.
  - Otherwise, when you compute the average, you will be dividing by zero.
- Your name should appear in the first line of your output.

Laboratory 5A

- Below is some sample output.

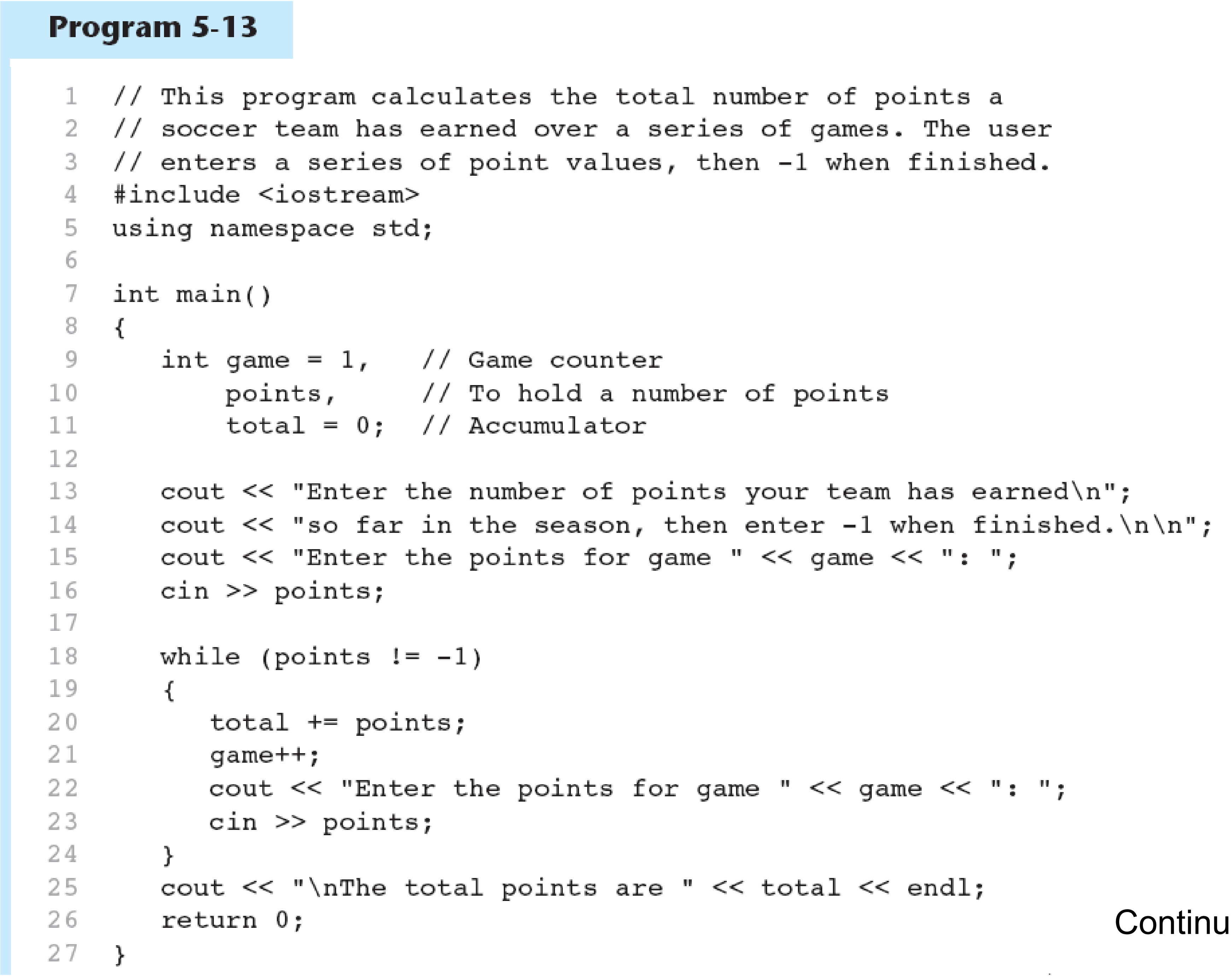


```
Jane Doe's Statistics Program
Enter a number: 82
Enter a number: 74
Enter a number: 16
Enter a number: 95
Enter a number: 47
Enter a number: 73
Enter a number: -99

The number of numbers entered is: 6
The sum of the numbers is: 387
The average value is:   64.5
```

Laboratory 5A

- Call your program: *YourName*-Lab05A-1.cpp
- If you are doing this Lab synchronously, demonstrate your solution to the instructor to receive proper credit.
- If you are doing this Lab asynchronously, submit your program to Canvas.
- Following are some programs in the text that might prove helpful.

A Sentinel in Program 5-13

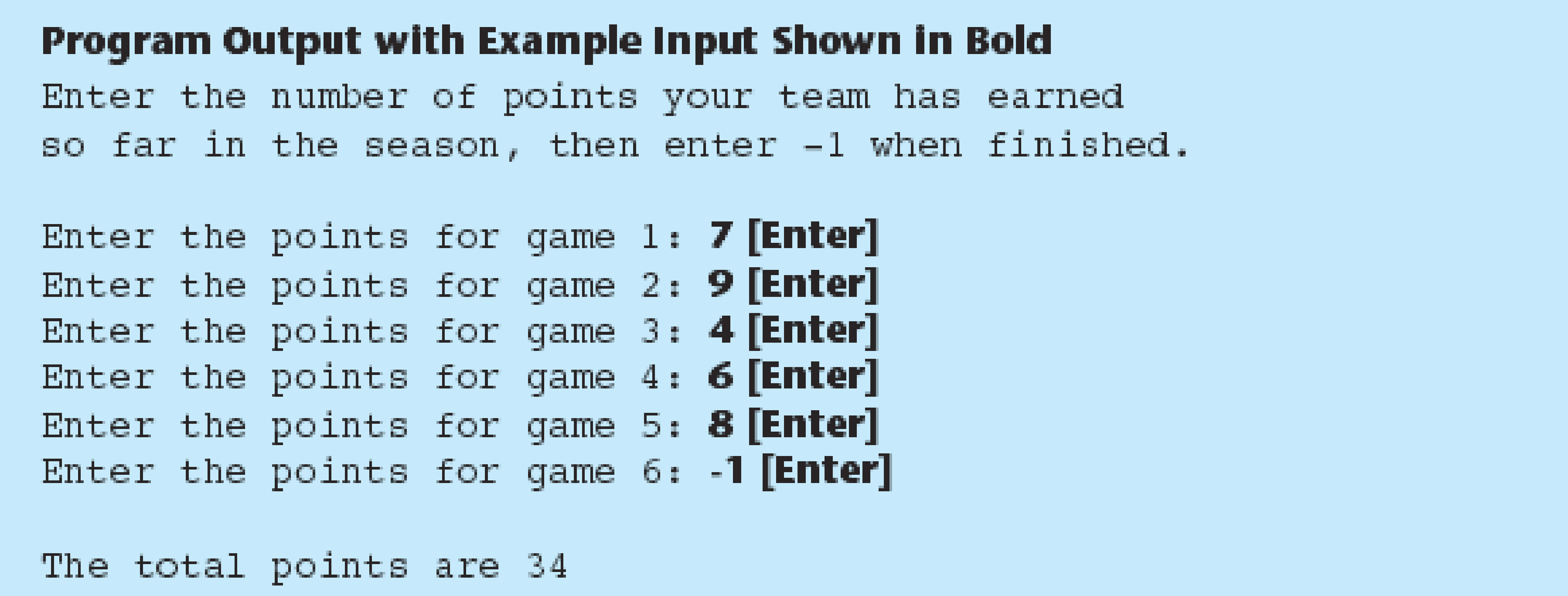


Program 5-13

```cpp
// This program calculates the total number of points a
// soccer team has earned over a series of games. The user
// enters a series of point values, then -1 when finished.
#include <iostream>
using namespace std;

int main()
{
   int game = 1,    // Game counter
       points,      // To hold a number of points
       total = 0;   // Accumulator

   cout << "Enter the number of points your team has earned\n";
   cout << "so far in the season, then enter -1 when finished.\n\n";
   cout << "Enter the points for game " << game << ": ";
   cin >> points;

   while (points != -1)
   {
      total += points;
      game++;
      cout << "Enter the points for game " << game << ": ";
      cin >> points;
   }
   cout << "\nThe total points are " << total << endl;
   return 0;
}
```

Continued…

A Sentinel in Program 5-13



Program Output with Example Input Shown in Bold

```
Enter the number of points your team has earned
so far in the season, then enter -1 when finished.

Enter the points for game 1: 7 [Enter]
Enter the points for game 2: 9 [Enter]
Enter the points for game 3: 4 [Enter]
Enter the points for game 4: 6 [Enter]
Enter the points for game 5: 8 [Enter]
Enter the points for game 6: -1 [Enter]

The total points are 34
```

Laboratory 5A – Part 2

- Now we’re going to repeat part A, but instead of using a numeric Sentinel value, we’re going to use an alpha character to end the loop.
- When that occurs, the program should print out:
  - The number of integers that you entered.
  - The sum of all the integers that you entered.
  - The average of all the integers that you entered.
- Your program should also test to be sure that the number of integers that you typed in is > 0.
  - Otherwise, when you compute the average, you will be dividing by zero.
- If you examine the sample program, Sum.cpp, you will see how this works, using **cin >> input** in the while loop condition.
- Call your program: *YourName*-Lab05A-2.cpp

Laboratory 5A – Part 2

- If you are doing this Lab synchronously, demonstrate your solution to the instructor to receive proper credit.
- If you are doing this Lab asynchronously, submit your program to Canvas.

Using the cin << Boolean result to end a loop of numeric data entry.

```
// Sum.cpp
#include <iostream>
using namespace std;
int main()
{
   cout << "Enter all the numbers you wish to Total, strike Q to quit." << endl;
   double total = 0;
   double input;
   while (cin >> input)
   {
      total = total + input;
   }
   cout << "The Total = " << total << endl;
   cin.clear();          // The following four statements will clear out the
   cin.ignore();         // false output from the while loop,
   cin.get();            // and get rid of the hanging key stroke, thus
   cin.get();            // enabling the output to be held on the screen.
   return 0;
}
```

Laboratory 5A – Part 3

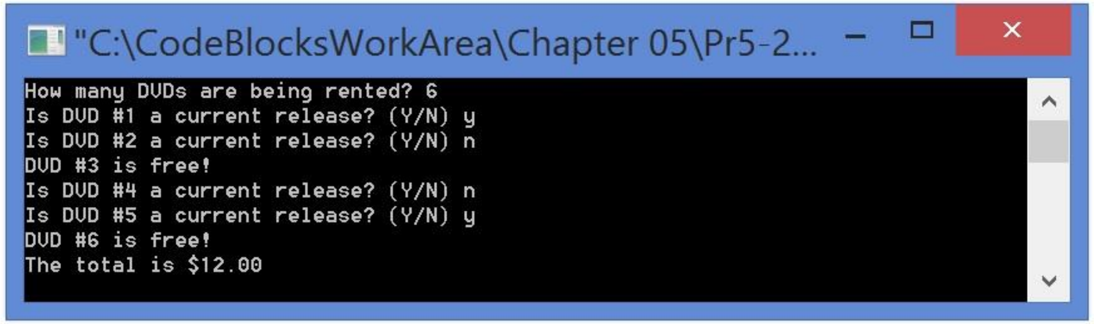
- In this Lab we are going to look at a C++ command which violates the rules of Structured Programming when used in loops, yet is tempting to use in certain situations.
- As the **continue** command is not necessary, we're going to show you where it might appear and how you can avoid its use altogether.
  - In Structured Programming, a loop is allowed to have one entrance and one exit.
  - The **continue** commands allows the loop to have more than one exit point.
  - This makes the loop more difficult to follow as well as debug.
- Program 5-26 shows the use of the **continue** command.
- Go ahead and enter this program and get it working. Test it so that you are clear about how it works.

Program 5-26: The Continue Statement - Example

```
// This program calculates the charges for DVD rentals.
// Every third DVD is free.
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
        int dvdCount = 1;    // DVD counter
        int numDVDs;         // Number of DVDs rented
         double total = 0.0;  // Accumulator
        char current;        // Current release, Y or N

        // Get the number of DVDs.
        cout << "How many DVDs are being rented? ";
        cin >> numDVDs;
```

Program 5-26 Concluded

```
    // Determine the charges.
     do
    {
              if ((dvdCount % 3) == 0)
              {
                       cout << "DVD #" << dvdCount << " is free!\n";
                       continue;  // Immediately start the next iteration
             }
             cout << "Is DVD #" << dvdCount;
             cout << " a current release? (Y/N) ";
              cin >> current;
              if (current == 'Y' || current == 'y')
                       total += 3.50;
             else
                       total += 2.50;
    } while (dvdCount++ < numDVDs);
    // Display the total.
    cout << fixed << showpoint << setprecision(2);
    cout << "The total is $" << total << endl;
     return 0;
}
```



```
How many DVDs are being rented? 6
Is DVD #1 a current release? (Y/N) y
Is DVD #2 a current release? (Y/N) n
DVD #3 is free!
Is DVD #4 a current release? (Y/N) n
Is DVD #5 a current release? (Y/N) y
DVD #6 is free!
The total is $12.00
```

Laboratory 5A – Part 3

- Program 5-26 shows the use of the **continue** statement in a C++ program.
- Your job is to rewrite the program so that the **continue** statement is no longer needed.
- Tip: You will need to change your outer **if ()** statement into an **if () else** statement.
- After you have worked out the new loop, rewrite the relevant lines and after you have it working, demonstrate your modified program to the Instructor.

Laboratory 5A, Part 3 Summary

- The take-away from this Lab is that in writing loops, the use of the **continue** command is unnecessary.
- With the **continue** command, there was no change in the number of lines for the loop when you wrote the loop avoiding its use.
- The result in avoiding this command is a loop with one exit, not two possible exits, hence a result that is easier to read as well as debug.
- Call your program: *YourName*-Lab05A-3.cpp
- If you are doing this Lab synchronously, demonstrate your solution to the instructor to receive proper credit.
- If you are doing this Lab asynchronously, submit your program to Canvas.

Lab 05 – Part 4

- **Problem 4:** Write a program that can be used as a math tutor for elementary school students.
- Display a menu to choose addition, subtraction, multiplication, or division.
- Generate two random numbers in the range of 1 to 12.
- For the subtraction problem, make sure the answer is positive (the larger number comes first).
- For the division problem, we only want integer answers. So multiply the two numbers to create the first number. For example, if the numbers generated were 4 and 5, instead of making the problem as: **4 / 5** (which has a decimal answer), multiply the numbers and use that as the first number, so the problem would be **20 / 5** (which has an integer answer).
- The program should loop as long as the user wants to continue. Use an input of ‘q’ or ‘Q’ to quit.

Test Case 1 for Part 4

Note Student Answers are in **bold**

```
Welcome to the Math Tutor!
Select an operation (+ - / *) or press q to quit:  +
9 + 10 = 19
That's right

Select an operation (+ - / *) or press q to quit:  -
12 - 8 = 2
The answer is 4

Select an operation (+ - / *) or press q to quit:  /
24 / 6 = 4
That's right

Select an operation (+ - / *) or press q to quit:  *
9 * 3 = 27
That's right

Select an operation (+ - / *) or press q to quit:  q
```

Laboratory 5A Part 4

- Call your program: *YourName*-Lab05A-4.cpp
- If you are doing this Lab synchronously, demonstrate your solution to the instructor to receive proper credit.
- If you are doing this Lab asynchronously, submit your program to Canvas.