

## LAB 5.1 Working with the **while** Loop

Copy and paste the following program into Visual Studio IDE.

---

```
// This program illustrates the use of a sentinel in a while loop.
// The user is asked for monthly rainfall totals until a sentinel
// value of -1 is entered. Then the total rainfall is displayed.

#include <iostream>
using namespace std;

void main()
{
    // Fill in the code to define and initialize to 1 the variable month

    float total = 0, rain;

    cout << "Enter the total rainfall for month " << month << endl;
    cout << "Enter -1 when you are finished" << endl;

    // Fill in the code to read in the value for rain

    // Fill in the code to start a while loop that iterates
    // while rain does not equal -1

    {
        // Fill in the code to update total by adding it to rain

        // Fill in the code to increment month by one

        cout << "Enter the total rainfall in inches for month " << month << endl;
        cout << "Enter -1 when you are finished" << endl;

        // Fill in the code to read in the value for rain

    }

    if (month == 1)
        cout << "No data has been entered" << endl;
    else
        cout << "The total rainfall for the " << month-1
            << " months is " << total << " inches." << endl;
}
```

---

**Exercise 1:** Complete the program above by filling in the code described in the statements in bold so that it will perform the indicated task.

**Exercise 2:** Run the program several times with various inputs including 0 for one or more months and -1.

**Print out the program and the running result for Exercise 2, and hand them in with the rest of the lab.**

**Note:** Try different input values including a 0 and other values. The last input should be -1.

## LAB 5.2 Working with the do-while Loop

Copy and paste the following program into Visual Studio IDE.

---

```
// This program displays a hot beverage menu and prompts the user to
// make a selection. A switch statement determines which item the user
// has chosen. A do-while loop repeats until the user selects item E
// from the menu.

#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    // Fill in the code to define an integer variable called number,
    // a floating point variable called cost,
    // and a character variable called beverage

    bool validBeverage;

    cout << fixed << showpoint << setprecision(2);

    do
    {
        cout << endl << endl;
        cout << "Hot Beverage Menu" << endl << endl;
        cout << "A: Coffee $1.00" << endl;
        cout << "B: Tea $ .75" << endl;
        cout << "C: Hot Chocolate $1.25" << endl;
        cout << "D: Cappuccino $2.50" << endl << endl << endl;
        cout << "Enter the beverage A,B,C, or D you desire" << endl;
        cout << "Enter E to exit the program" << endl << endl;

        // Fill in the code to read in beverage

        switch ( beverage )
        {
            case 'a':
            case 'A':
            case 'b':
            case 'B':
            case 'c':
            case 'C':
            case 'd':
            case 'D': validBeverage = true;
                     break;
            default: validBeverage = false;
        }
    }
```

```

if (validBeverage == true)
{
    cout << "How many cups would you like?" << endl;

    // Fill in the code to read in number
}

// Fill in the code to begin a switch statement
// that is controlled by beverage

{
    case 'a':
    case 'A': cost = number * 1.0;
              cout << "The total cost is $ " << cost << endl;
              break;

    // Fill in the code to give the case for tea ( $0.75 a cup)
    // Fill in the code to give the case for hot chocolate ($1.25 a cup)
    // Fill in the code to give the case for cappuccino ($2.50 a cup)

    case 'e':
    case 'E': cout << " Please come again" << endl;
              break;
    default:cout << // Fill in the code to write a message
                  // indicating an invalid selection.

              cout << " Try again please" << endl;
}

} // Fill in the code to finish the do-while statement with the
  // condition that beverage does not equal E or e.

// Fill in the appropriate return statement
}

```

**Exercise 1:** Fill in the indicated code to complete the above program. Then compile and run the program several times with various inputs. Try **ALL** the possible relevant cases (input: A, B, C, D, E, a, b, c, d, e or other invalid letters), and check your results.

**Print out the program and the running result for Exercise 1, and hand them in with the rest of the lab.**

**Note:** Try different input values, including two valid inputs and one invalid input.

## LAB 5.3 Working with the **for** Loop

Copy and paste the following program into Visual Studio IDE.

---

```
// This program has the user input a number n and then finds the
// mean of the first n positive integers

#include <iostream>
using namespace std;

int main()
{
    int value;           // value is some positive number n
    int total = 0;       // total holds the sum of the first n positive numbers
    int number;          // the amount of numbers
    float mean;          // the average of the first n positive numbers

    do
    {
        cout << "Please enter a positive integer" << endl;
        cin >> value;
    } while (value <= 0);

    // First find the running total from 1 to value.
    for (number = 1; number <= value; number++)
    {
        total += number;
    } // curly braces are optional since there is only one statement

    // Then find the mean. Note the use of the typecast operator here.
    mean = static_cast<float>(total) / value;

    cout << "The mean average of the first " << value
         << " positive integers is " << mean << endl;

    return 0;
}
```

---

**Exercise 1:** Why is the typecast operator needed to compute the mean in the statement `mean=static_cast(float)(total)/value;`? What do you think will happen if it is removed? Modify the code and try it. Record what happens. Make sure that you try both even and odd cases. Now put `static_cast<float> total` back in the program.

**Exercise 2:** Modify the code so that it computes the mean of the consecutive positive integers  $n, n+1, n+2, \dots, m$ , where the user chooses  $n$  and  $m$ . For example, if the user enters 3 and 9, then the program should find the mean of 3, 4, 5, 6, 7, 8, and 9, which is 6.

**Print out the program and the running result for Exercise 2, and hand them in with the rest of the lab.**

**Note:** Try different input values, including one invalid input value and one valid input value.