Lab 5

LAB 5.1 Working with the `while` Loop

Bring in program `while.cpp` from the Lab 5 folder. The code is shown below:

```
// PLACE YOUR NAME HERE

#include <iostream>
using namespace std;

int main()
{
   char letter = 'a';

   while (letter != 'x')
   {
      cout << "Please enter a letter" << endl;
      cin >> letter;

      cout << "The letter you entered is " << letter << endl;
   }

   return 0;
}
```

Exercise 1
This program is not user friendly. Run it a few times and explain why.

Exercise 2
Add to the code so that the program is more user friendly.

Exercise 3
How would this code affect the execution of the program if the `while` loop is replaced by a `do-while` loop? Try it and see.

Bring in program `sentinel.cpp` from the Lab 5 Folder. The code is shown below:

```
// 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.

// PLACE YOUR NAME HERE

#include <iostream>
using namespace std;

int 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;

   return 0;
}
```

Exercise 4

Complete the program above by filling in the code described in the statements so that it will perform the indicated task.

Exercise 5

Run the program several times with various input. Record your results. Are they correct? What happens if you enter –1 first? What happens if you enter only values of 0 for one or more months? Is there any numerical data that you should not enter?

Exercise 6

What is the purpose of the following code in the program above?

```
if (month == 1)
   cout << "No data has been entered" << endl;
```

LAB 5.2 Working with the `do-while` Loop

Bring in the program `dowhile.cpp` from the Lab 5 folder. The code is shown below:

```
// 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.

// PLACE YOUR NAME HERE

#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 hot chocolate ($1.25 a cup)

            // Fill in the code to give the case for tea ( $0.75 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 and record your results.

Exercise 2

What do you think will happen if you do not enter A, B, C, D or E? Try running the program and inputting another letter.

Exercise 3

Replace the line

```
if (validBeverage == true)
```

with the line

```
if (validBeverage)
```

and run the program again. Are there any differences in the execution ofthe program? Why or why not?

LAB 5.3 Working with the `for` Loop

Bring in program `for.cpp` from the Lab 5 folder. This program has the user input a number *n* and then finds the mean of the first *n* positive integers. The code is shown below:

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

// PLACE YOUR NAME HERE

#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

   cout << "Please enter a positive integer" << endl;
   cin >> value;

   if (value > 0)
   {
      for (number = 1; number <= value; number++)
      {
         total = total + number;
      }   // curly braces are optional since there is only one statement

      mean = static_cast<float>(total) / value;   // note the use of the typecast
                                                  // operator here
      cout << "The mean average of the first " << value
          << " positive integers is " << mean << endl;
   }
   else
      cout << "Invalid input - integer must be positive" << 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)` back in the program.

Exercise 2

What happens if you enter a float such as 2.99 instead of an integer for `value` ? Try it and record the results.

Exercise 3

Modify the code so that it computes the mean of the consecutive positive integers *n, n+1, n+2, . . . , m*, where the user chooses *n* and *m*.

For example, if the user picks 3 and 9, then the program should find the mean of 3, 4, 5, 6, 7, 8, and 9, which is 6.

LAB 5.4 Nested Loop

Bring in program `nested.cpp` from the Lab 5 folder. The code is shown below:

```
// This program finds the average time spent programming by a student
// each day over a three day period.

// PLACE YOUR NAME HERE

#include <iostream>
using namespace std;

int main()
{
   int numStudents;
   float numHours, total, average;
   int student, day = 0;    // these are the counters for the loops

   cout << "This program will find the average number of hours a day"
       << " that a student spent programming over a long weekend\n\n";
   cout << "How many students are there ?" << endl << endl;
   cin >> numStudents;

   for (student = 1; student <= numStudents; student++)
   {
      total = 0;

      for (day = 1; day <= 3; day++)
      {
         cout << "Please enter the number of hours worked by student "
             << student << " on day " << day << "." << endl;
         cin >> numHours;

         total = total + numHours;
      }

      average = total / 3;

      cout << endl;
      cout << "The average number of hours per day spent programming by "
          << "student " << student << " is " << average
          << endl << endl << endl;
   }

   return 0;
}
```

Exercise 1

Note that the inner loop of this program is always executed exactly three times—once for each day of the long weekend. Modify the code so that the inner loop iterates *n* times, where *n* is a positive integer input by the user. In other words, let the user decide how many days to consider just as they choose how many students to consider.

Sample Run:

```
This program will find the average number of hours a day that a student spent programming
over a long weekend

How many students are there ?
2
Enter the number of days in the long weekend
2

Please enter the number of hours worked by student 1 on day 1.
4
Please enter the number of hours worked by student 1 on day 2.
6

The average number of hours per day spent programming by student 1 is 5


Please enter the number of hours worked by student 2 on day 1.
9
Please enter the number of hours worked by student 2 on day 2.
13

The average number of hours per day spent programming by student 2 is 11
```

Exercise 2

Modify the program from Exercise 1 so that it also finds the average number of hours per day that a given student studies biology as well as programming. For each given student include two prompts, one for each subject. Have the program print out which subject the student, on average, spent the most time on.

LAB 5.5 Reading and Writing to a File

Bring in `billfile.cpp` from the Lab 5 folder. The code is as follows:

```
// This program will read in the quantity of a particular item and its price.
// It will then print out the total price.
// The input will come from a data file and the output will go to
// an output file.

// PLACE YOUR NAME HERE

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

int main()
{
   ifstream dataIn;   // defines an input stream for a data file
   ofstream dataOut;    // defines an output stream for an output file
   int quantity;      // contains the amount of items purchased
   float itemPrice;   // contains the price of each item
   float   totalBill;   // contains the total bill, i.e. the price of all items
   dataIn.open("transaction.dat");   // This opens the file.
   dataOut.open("bill.out");

   // Fill in the appropriate code in the blank below

   _________ << setprecision(2) << fixed << showpoint;   // formatted output



   // Fill in the input statement that brings in the
   // quantity and price of the item

   // Fill in the assignment statement that determines the total bill.

   // Fill in the output statement that prints the total bill, with a label,
   // to an output file

   return 0;
}
```

Exercise 1

This program gets the information from a file and places the output to a file. You must create the data file. Your instructor will tell you how to create the data file and where to put it. Create a data file called `transaction.dat` that has the following information:

22
10.98

Exercise 2

Fill in the blank and the statements that will read the data from the
file and print the following to `bill.out` :

```
The total bill is $241.56
```

LAB 5.6 Student Generated Code Assignments

Option 1: Write a program to display all the prime numbers within a range of integers.

Sample Run 1:

```
Input a range of integers.
From integer a ( a must be greater than 1 ) :
0
From integer a ( a must be greater than 1 ) :
-1
From integer a ( a must be greater than 1 ) :
2
To integer b ( b must be greater than or equal to a ) :
1
To integer b ( b must be greater than or equal to a ) :
100
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
25 prime numbers.
```

Sample Run 2:

```
Input a range of integers.
From integer a ( a must be greater than 1 ) :
2
To integer b ( b must be greater than or equal to a ) :
400
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113
127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241
251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383
389 397
78 prime numbers.
```

Option 2: Write a program that performs a survey tally on beverages. The program should prompt for the next person until a sentinel value of –1 is entered to terminate the program. Each person participating in the survey should choose their favorite beverage from the following list:

1. Coffee
2. Tea
3. Coke
4. Orange Juice

Sample Run:

```
Please input your choice of drink (a number from 1 to 4
1  -  Coffee
2  -  Tea
3  -  Coke
4  -  Orange Juice
-1 -  QUIT
4

Please input your choice of drink (a number from 1 to 4
1  -  Coffee
2  -  Tea
3  -  Coke
4  -  Orange Juice
-1 -  QUIT
1

Please input your choice of drink (a number from 1 to 4
1  -  Coffee
2  -  Tea
3  -  Coke
4  -  Orange Juice
-1 -  QUIT
3

Please input your choice of drink (a number from 1 to 4
1  -  Coffee
2  -  Tea
3  -  Coke
4  -  Orange Juice
-1 -  QUIT
1

Please input your choice of drink (a number from 1 to 4
1  -  Coffee
2  -  Tea
3  -  Coke
4  -  Orange Juice
-1 -  QUIT
1

Please input your choice of drink (a number from 1 to 4
1  -  Coffee
2  -  Tea
3  -  Coke
4  -  Orange Juice
-1 -  QUIT
-1


The total number of people surveyed is 5
The results are as follows:

Beverage   Number of Votes
Coffee        3
Tea           0
Coke          1
Orange Juice  1
```

Option 3: Suppose Dave drops a watermelon off a high bridge and lets it fall until it hits the water. If we neglect air resistance, then the distance d in meters fallen by the watermelon after t seconds is d = $0.5 * g * t^2$, where the acceleration of gravity g = 9.8 meters/second ^{2} . Write a program that asks the user to input the number of seconds that the watermelon falls and the height *h* of the bridge above the water. The program should then calculate the distance fallen for each second from *t* = 0 until the value of *t* input by the user. If the total distance fallen is greater than the height of the bridge, then the program should tell the user that the distance fallen is not valid.

Sample Run 1:

```
Please input the time of fall in seconds:
2
Please input the height of the bridge in meters:
100


Time Falling (seconds)    Distance Fallen (meters)
**********************    ************************
                     0                           0
                     1                         4.9
                     2                        19.6
```

Sample Run 2:

```
Please input the time of fall in seconds:
4
Please input the height of the bridge in meters:
50


Time Falling (seconds)    Distance Fallen (meters)
**********************    ************************
                     0                           0
                     1                         4.9
                     2                        19.6
                     3                        44.1
                     4                        78.4

WARNING-Bad Data:  The distance fallen exceeds the height of the bridge
```