

Lab 7 - CMPS 1044 - Computer Science I

While Loops

Objectives: Demonstrate use of while loops in a variety of scenarios:

- 1) Counting loop
- 2) Sentinel loop
- 3) Check for specific condition

WHILE: The while loop is a programming construct which executes a set of statements repeatedly while a stated condition is true.

```
while (condition)    // NOTE: no semicolon here
{
    Statement 1;    // Place one or more statements between the
                  //brackets.
    ...            // Statements between brackets are referred to as
    Statement n;    // the BODY of the loop
}
```

Execution: The (condition) is tested.

- If it is false, program execution skips all commands in the brackets and continues with the instructions following the closing bracket
- If it is true
 - Execute the statements in the **body** of the loop
 - Return to the **while** statement, retest the condition, and repeat these steps

Notes on use of While Loop:

- Variables in (condition) must be **initialized** prior to the **while** statement
- **At least one** of the variables in the (condition) **MUST** change in the body of the loop, or an infinite loop will occur.

WHILE as a counting loop – Used to perform the body of the loop a pre-determined number of times. The following loop will execute the body of the loop 10 times. Type in and execute the following code.

```
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
int main()
{
    int Count = 1;
    while (Count <= 10)    // NOTE: no semicolon here
    {
        cout << "Count is " << Count << '\n';
        Count += 1;        // increment counter until it reaches 11
    }
    cout <<"Outside the loop now. Count has final value of "<<Count<< '\n';
```

```

    system("pause");
    return 0;
}

```

Consider the following problem: Enter 7 temperatures from the keyboard – each one representing a daily temperature for the previous week. Sum the 7 values then calculate and print the average temperature for the week. Modify the program above to solve this problem. Before you do, consider the following issues:

- 1) What variables will you need for input and output?
- 2) Which operations will you perform one time (outside the body of the loop), and which operations are performed 7 times (inside the body of the loop)?

Now, modify the program above to solve the problem. Execute with the following 7 data items:
82 75 61 73 89 90 69 -- Average is 77

WHILE checking for a sentinel – A sentinel is a data value that marks the end of a data set. Usually it is a value that is different from the legitimate data values. If you have a rather large data set which changes in size over time, it may be easier to place a **sentinel** value at the end of the data set to indicate the end rather than trying to count the number of data items and use a counting loop.

Example: Suppose the data values are person's ages. Legitimate data values are positive values so a sentinel value could be a negative value. The sentinel value is usually ignored, not processed.

Statement of Problem: Read and process a set of values UNTIL a negative value is read.

Sample Data Set: The following data set represents 8 valid data items followed by a sentinel.

23 45 53 29 16 85 23 66 -25

Consider the following problem: Read and sum a set of integers until a negative value is read. Do not include the negative value (sentinel) in the sum. Type in and execute the program.

```

int DataVal;
int Sum = 0;           // initialize sum variable
cin >> DataVal;       // read the first data item
while (DataVal >= 0)  // check for a negative value
{
    Sum += DataVal;
    cin >> DataVal;    // read data item to update condition
}
cout << "Outside the loop now. The sum of the values is " << Sum <<
'\n';

```

Now modify the above code to calculate the average of the numbers you are reading. You MUST make the code count the number of values for the division. You CANNOT count by hand and type in the number.

WHILE checking for a specific condition – A while loop can execute until any stated condition is reached.

Examples:

1. Read integers as input stopping when the total value of integers input is greater than 100.
2. Read characters as input. Repeat as long as the user enters 'Y' or 'y' but stops with any other character.
3. Ask the user to enter a 1, 2, 3, or 4. Repeat as long as they do; Stop if any other value is entered.
4. Just about any other condition you can think of!

Type in and execute the following solution to Example 1.

```
int DataVal;
int Sum = 0;           // initialize sum variable
cin >> DataVal;       // read the first data item
Sum += DataVal;       // Add data item to the sum
while (Sum <= 100)    // check for a negative value
{
    cin >> DataVal;    // read data item to update condition
    Sum += DataVal;    // Add data item to the sum
}
cout << "Outside the loop now. The final sum of the values is " << Sum <<
'\n';
```

Modify the above program to print out each value as it is read, count the number of values read, and print it out with the total.

This week's assignment is in the form of a word problem. You should use this word problem when you are writing a description of what the program does in your header. Be sure to add three comments to the body of your program and print your header and results to an output file.

ASSIGNMENT: Joe just graduated college and has been offered his first real job as a widget designer at Widgets USA, Inc. He is disappointed in the offered beginning salary, only \$20,000 per year. But he is promised a 10% raise each year. Other job opportunities offer more to start, but have no promised raises. To help Joe decide, print out a table of his new salary each year until his salary reaches or exceeds \$50,000, including the last year and final salary. Print an appropriate statement about how many years this will take. The table is to look as follows:

Year	Salary
1	\$ 20000.00
2	\$ 22000.00