**Lab Experience 2**

**Objectives:**

1. To learn interactive programming and to format the information produced by the program.

2. To be able to convert arithmetic expressions into valid C++ mathematical operations

3. To work with constants and mathematical functions found in the cmath header file.

4. To understand explicit and implicit type coercion.

**Background Information:**

All input/output in C++ is considered a stream of data/information that is routed to/from the output/input device that is specified by the object. cin is the object that specifies the input stream is coming from the keyboard and cout is the output stream that specifies the output is being sent to the monitor/console. Items are either inserted onto the output stream or extracted from the input stream. Each stream is associated with an input/output buffer (a segment of memory) and the buffer is associated with an appropriate peripheral device. Some common peripheral devices are the computer monitor, keyboard, and internal/external drives.

This lab covers topics discussed in Chapters 2 and 3 of your textbook.

**Input Instructions**

The cin object is used to transfer information from the keyboard into the declared identifiers (variables). Everything typed at the keyboard is in ASCII format, hence it is a character. The extraction operator, >>, is used to convert the input data into the data type specified by the identifier.

For example: cin >> purchasePrice;

This statement will extract characters from the data stream until it encounters a character that does not match the data type of purchasePrice. Suppose purchasePrice is of type int, all whole numbers will be extracted until a character that is not a whole number is encountered. This could be a letter, decimal point, special character, or whitespace.

All of the characters that are integer will now be converted from their ASCII representation into their binary integer representation. A cout statement is used to precede all cin statements since this is interactive input and the user needs directions on what type of data to input.

**Data Type Conversions/Coercions**

When mixing data types in expressions, the computer will automatically convert the data into another data type as specified by the promotion/demotion rules found on pages 98-99 in section 3.3.

There are two types of data conversion in C++:

**Implicit** --- The compiler converts the data type based upon the rules found on pages 98 and 99. A warning message can appear stating loss of data can happen. If this is ok, ignore the warning message, if it is not ok, then fix the statement to cause the warning message to disappear.

**Explicit** --- The programmer places the necessary statements in the program to temporarily convert the variables data type into another data type. This is accomplished with static\_cast statement. The syntax is:

static\_cast<data type to convert to>(value to be converted temporarily)

**Example:**

int number = 5;

double dbl;

dbl = number; // implicit conversion will take place. The value

// in dbl will be 5.0 no warning msg

number = static\_cast<int>(dbl); // explicit conversion will take place.

// The decimal value contained in dbl will

// be truncated to the whole number. That

// is, if dbl contained 5.95 only the

// number 5 would be stored in number.

**Lab Exercises**

**Problem 1: Fill in the blank:**

Answer the following questions based upon material presented in lecture and found in chapters 2 and 3 of your textbook.

1. What is the final value (evaluated in C++) of the following expression?

(5 – 16 /2 \* 3 + (3 + 2 / 4)) – 5 Answer: \_\_\_\_\_\_\_-20.5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Write the following expression in C++ format.

C++ Expression: \_\_\_\_\_E=(m(c\*c)/sqrroot(1-(v\*v)/(c\*c))\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. What is the preprocessor directive that must be included for cin and cout to be used in a C++ program?

Answer: \_\_\_\_\_\_\_\_#include <iostream>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Blank spaces, tabs, and newlines in an input line or data file are referred to as \_\_\_\_\_\_\_\_\_White space\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

5. The << in a cout statement is called the \_\_\_\_\_\_insertion\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator.

6. The #include<\_\_\_\_\_\_\_\_iostream\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_> directive is needed for setw and setprecision.

7. The ‘\n’ is a special character called \_\_\_\_\_\_escape\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ character.

8. The >> in a cin statement is called the \_\_\_\_\_\_extraction\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator.

**Problem 2:**

1. Start Visual Studio.
2. Create a new project solution and add a new C++ file called: lab2prob2.cpp.
3. Type the program listed below.

// Written By Tim Wrenn

// Class: Computer Science 1106

// Lab 2 Program

// Program to convert Fahrenheit temperature to centigrade

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

double tempCentigrade, // temperature in centigrade

tempFahrenheit; // temperature in Fahrenheit

// Request input from user

cout<< "\nPlease enter the temperature in Fahrenheit that "

<< "you wish to convert: ";

cin >> tempFahrenheit;

// Perform conversion

tempCentigrade = 5/9 \* (tempFahrenheit - 32);

// Display results

cout << "\n\n" << tempFahrenheit << " degrees Fahrenheit is equal to ";

cout << fixed << showpoint << setprecision(1);

cout << tempCentigrade << " degrees centigrade.\n\n";

return 0;

}

Answer the following questions by using the program above.

1. Fix all syntax errors (if there are any).
2. Execute the program using the values in the tempFahrenheit column as input into your program and recording your results under the tempCentigrade column. Use a calculator to perform the same calculation. This is called white box testing.

|  |  |  |
| --- | --- | --- |
| tempFahrenheit | tempCentigrade | Calculator Result |
| 5 | 0 | -15 |
| 32 | 0 | 0 |
| -5 | 0 | -20.555555555555 |
| 30 | 0 | -1.1111111111111 |

Does the calculator result match the result from the program? \_\_\_\_No

Change the equation tempCentigrade = 5/9 \* (tempFahrenheit - 32); to

tempCentigrade = (tempFahrenheit - 32) \* 5/9 ;

and re-run your program and record your results below:

|  |  |  |
| --- | --- | --- |
| tempFahrenheit | tempCentigrade | Calculator Result |
| 5 | -15 | -15 |
| 32 | 0 | 0 |
| -5 | -20.6 | -20.55555555555 |
| 30 | -1.1 | -1.111111111111 |

Does the calculator result match the result from the program? \_\_\_\_\_Yes\_

Why does the calculator result match your program’s results? Explain your answer using C++ terminology and Computer Science terminology.

The division in the first statement divided five by 9 times the difference of the tempFahrenheit and 32; This would result in 5 being divided by a large number, which would lead to a small double number; Because the values where in integers instead of float form with decimals, then it performed integer division and dropped off the values after 0. This resulted in all the test values being 0. In the second statement, the product of 5 times the difference of the tempFahrenheit and 32 is divided by nine, which uses integer division again, but the product is divided by 9, the number won't simply shorten to 0.

**Due Dates:** According to the due date posted for the drop box folder.

**What to hand in:**

1. Save the word document using your name and the lab number, i.e. timwrennlab2.docx
2. Place the word document into the Lab Two dropbox folder in D2L.
3. Hand in a print out of your word document.