CIS121 Laboratory Three

Objectives of this lab:

1. Introduce three of the primitive data types in C++.
2. Discuss some of the elementary operations that can be performed on these types.
3. Investigate how to read input from the keyboard.

Part 1: Primitive Data Types

Intuitively, something is different between data consisting of numeric values (such food prices at McDonald's or the number of tacos you can get for a buck at Taco Bell) and data consisting of strings of characters (such as the names of your best friends or the lyrics to your favorite song). In one case we can meaningfully apply mathematical operations such as addition and subtraction, while in the other we cannot. In programming terminology we say that the two groups of data are of different type.

The C++ programming language accounts for several elementary types of data. All other types of data are constructed as conglomerates of these primitive types. Our goal for now is to learn the fundamentals of three basic types: **integer, real, and character**.

The integer data type encompasses the counting numbers and their negatives:

. . . -4, -3, -2, -1, 0, 1, 2, 3, 4, . . .

The data type character accounts for data consisting of a single symbol such as a letter of the alphabet, a punctuation mark, or any special symbol, such as $, %, and @. Data items of type character are normally stored using ASCII code in C++ applications. In future sessions you will learn how words and sentences can be constructed as collections of data of type character.

Variables

A variable is a name used in a program to refer to an item of data. Such names, also called identifiers, consist of any combination of letters, digits, and underscores, as long as they don't begin with a digit. Thus, Ritz_bits, X25, and Didi would be valid variable names, whereas 7eleven, num miles, and 409 would not be. Good programming practice promotes the use of identifiers that reflect their role within the program. For example, a variable referring to temperature values might be called temperature or degrees.

Before a variable name can be used in a program it must be declared; that is, the particular name along with its data type must be stated. This is done via a variable declaration statement having the form

data_type variable_list;

where *data_type* indicates the type associated with the variables being declared and *variable_list* is a list of variable names separated by commas. Keywords are used to represent the data types. These words have strict, preassigned meanings and cannot be used for other purposes (such as names of variables) elsewhere in a program.

```
int servings, fat, calories;
float sodium;
char ingredients;
```

establish servings, fat, and calories as variables of type int; sodium as a variable of type float; and ingredients as a variable of type char.

The language C++ allows a value to be assigned to a variable at the time the variable is declared. Such initialization of variables is done by following the variable name by an equal sign (=) and the value to be assigned. Thus, the statements

```
int x = 1, y = 2;
float z = 3.75;
char sym = 'a';
```

not only establish x, y, z, and sym as variables of type int, int, float, and char, respectively, but assign the variables the starting values 1, 2, 3.75, and a. (Note that data of type character is enclosed with single quotation marks when it appears within a program.)

Reading and Writing Data of Primitive Types

When you communicate with the world outside of your C++ program you use something called an I/O stream. (I/O is a commonly used abbreviation for Input/Output). An I/O stream is a sequence of characters that can be either directed into or out of your program. As you learned in the previous laboratory session, the object cout can be used to display messages on the monitor screen. cout is the stream that is associated with the screen output device. Recall that the operator << was used to display information on the screen.

You can also use the << operator to have cout display values that are stored in variables in your program. For example, consider the second line in the sequence

```
cout << "One gallon of ice cream = ";
cout << scps_gal;
cout << " scoops.\n";
```

If the variable scps_gal is assigned the value 103, then the statement sequence causes the message

```
One gallon of ice cream = 103 scoops.
```

to appear on the monitor screen.

The << operator can be used multiple times in a single statement to display multiple items of data. Thus, assuming that the variable v, of type char, is assigned the character A and the variable p, of type int, is assigned the value 75, the statement

```
cout << "U.S.RDA: vitamin " << v << " = " << p << " percent.\n";
```

produces the line

```
U.S.RDA: vitamin A = 75 percent.
```

Whereas the << operator is used to send data to the monitor screen via output stream cout, the >> operator is used to receive data from the keyboard. The input stream from the keyboard is known as cin. Thus the statement

```
cin >> scps_gal;
```

retrieves the value typed at the keyboard and stores it in the variable scps_gal

We can use the >> operator multiple times in a single statement to receive several values from the keyboard. For example, if sym is of type char and num is of type int, then

```
cin >> sym >> num;
```

would tell cin to read two pieces of information from the keyboard and assign the first to the variable sym and the second to num

When the object cin is used, characters representing the keys typed at the keyboard are collected in the input stream until a new line marker (generated by typing the Enter key) appears in the input stream. At this time, cin processes the contents of the stream by putting values into variables as dictated by the use of the >> operator. If all of the >> operators used can be satisfied by the data in the input stream, cin makes the appropriate variable assignments and terminates activities; otherwise, cin must wait for more data to be entered at the keyboard, followed by another new line marker. Thus, the statement

```
cin >> sym >> num;
```

will cause the program's execution to pause until cin has assigned values to both variables.

For now, suppose that age is a variable of type int and sym is a variable of type char. If the keys 1, 2, 5, and Enter are typed in response to the command

```
cin >> age >> sym;
```

then cin will assign the value 125 to the variable age and the program's execution will seem to be stuck as cin waits for another character to be entered.

Experiment 3.1

Execute the following program, and record the results below.

```
#include <iostream>
using namespace std;
int main()
{       int num = 2;
        char sym_1 = 'c';
        char sym_2 = 'i';
        char sym_3 = 'n';
        cout<<"Give me "<< num << " scoops of \n";
        cout<<sym_1<<"ho"<<sym_1<<"olate ";
        cout<<sym_1<<"ho"<<sym_1<<"olate "<<sym_1<<"hip ";
        cout<<sym_2<<sym_3<<" a waffle cone please!\n";
        return 0;
}
```

Experiment 3.2

1) Execute the following program. In response to the first request, type the character a followed by the Enter key. In response to the second request, type bcd, followed by the Enter key.

   What was left in the input stream after executing cin the first time?

```
#include <iostream>
using namespace std;
int main()
{
        char a, b, c;
        cout << "Enter a character.\n";
        cin >> a;
        cout << "\nThe character entered was " << a << ".\n";
        cout << "Enter three more characters." << endl;
        cin >> a >> b >> c;
        cout << "\nThe characters entered were " << a << b << c <<".\n";
        return 0;

}
```

2) Execute the program again. This time answer the first request by typing efgh followed by W. Explain the results.

Experiment 3.3

1) Execute the following program. In response to the first request, type two integers followed by the Enter key. In response to the second request, type two integers followed by the Enter key. Explain the results.

```
#include <iostream>

using namespace std;
int main()
{
 int a, b, c;
 cout << "Enter an integer.\n";
 cin >> a;
 cout << "\nThe value entered was " << a << endl;
 cout << "Enter two integers.\n";
 cin >> b >> c;
 cout << "\nThe values entered were "<<b<<" and "<<c << endl;
 return 0;
}
```

2) Execute the program in Step 1 again. In response to the first request, type a single integer followed by the Enter key. In response to the second request, type a single integer, followed by the Enter key, followed by a single integer, followed by the Enter key. Explain the results.

Part 2: Elementary Operations and the Assignment Statement

In addition to using cin, values can be assigned to variables by means of assignment statements. The statement

```
pizza = 14.50;
```

requests that the floating-point value 14.50 be assigned to the variable pizza. In general, the assignment statement is used to assign the value found on the right of the = symbol to the variable found on the left of the = symbol.

The right side of an assignment statement can be any expression whose value is compatible with the variable on the left side. Thus, assuming that calories and fat are variables of type integer, the following assignment statements are valid.

```
calories = 270;
fat = calories;
```

In the case of numeric data, the traditional arithmetic operations can be used in an expression on the right side of the assignment statement to direct the computation of the value to be assigned. Here the symbols +, –, *, and / are used to represent addition, subtraction, multiplication, and division, respectively. Thus, the statement

```
pounds = calories * fat;
```

assigns the product of calories and fat to pounds.

Parentheses can be used to clarify the order in which the expression on the right side of an assignment statement is to be evaluated. Thus,

```
cal_serving = (fat_calories + sugar_calories) / servings;
```

would correctly assign the true value to cal_serving, whereas

```
cal_serving =  fat_calories + sugar_calories / servings;
```

would not.

Experiment 3.4

Execute the following program and explain the results. (What can be said about the value of a variable that has not been assigned a value?)

```
#include <iostream>
using namespace std;
int main()
{
        int x, y;
        cout << "x = " << x << endl;
        cout << "y = " << y << endl;
        y = 25;
        cout << "x = " << x << endl;
        cout << "y = " << y << endl;
        return 0;
}
```

Experiment 3.5

Execute the following program and explain the results and the usage of the division operator (/).

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

```
int main()
{
  int integer_result;
  float float_result;
  integer_result = 7/3;
  float_result = 7/3;
  cout << integer_result << endl;
  cout << float_result << endl;
  integer_result = 12.6/3;
  float_result = 12.6/3;
  cout << integer_result << endl;
  cout << float_result << endl;
  return 0;
}
```

Part 3: Programming Exercises

Complete the following questions.

1. Compute the Area and Volume of a Cylinder (required)

We are computing the surface area and the volume of a cylinder. Write a program to compute the volume V of a cylinder of radius r and height h. The program will ask the user for the radius of the cylinder, and the height of the cylinder.

The following equation is used to compute the area and the volume of a cylinder:

area_cylinder = $2\pi r h + 2\pi r^2$

volume_cylinder = $\pi r^2 h$

Example Output:

```
Lab 3 - Exercise #1 by John Smith

Press return after entering a number.
Enter the radius of the cylinder (meters):
2.5
Enter the height of the cylinder (meters):
8.64

The surface area of a cylinder of radius 2.5 and height 8.64 is 174.987 square
meters (or m^2)
The volume of a cylinder of radius 2.5 and height 8.64 is 169.646 cubic meters
(or m^3)
```

Required Test Cases:

```
radius = 5.12 m, height = 6.8 m
radius = 5.12 m, height = 6.8 m
```

2. **Stock Commission (required)**

Susan bought 750 shares of stock at a price of $35.00 per share. She must pay her stockbroker a 2 percent commission for the transaction. Write a program that calculates and displays the following:

- The amount paid for the stock alone (without the commission)
- The amount of the commission
- The total amount paid (for the stock plus the commission)

Your program should ask the user to enter the number of shares and amount of the commission. Calculate and display the total amount paid.

Sample Output:

How many shares did you buy? 750
How much is each share? 35.00
Stock: $26250
Commission: $525
Total: $26775
750 shares of stock at a price of $35.00 per share, the total is 26775

Press any key to continue…

Submission

1. Submit sources code and screenshot of the output for Part 3- Programming Exercises.
2. Use the header format template posted on D2L.
3. Add line comments
4. [Recommended] Send a lab report which includes a summary about your lab experiments 3.1 – 3.5.