LAB 04: Variables and Data types

Objective(s):
Understanding of Data types, Variables and operations that can be performed using them.
CLOs: CLO1, CLO2
Data types:
Data types portray the amount of memory to be allocated. Most common data types are integer, floating- point, character, strings, bool.
Built – In Data Types in C++:

Type Name	Bytes	Other Names	Range of Values
Int	4	signed	-2,147,483,648 to 2,147,483,647
unsigned int	4	unsigned	0 to 4,294,967,295
__int8	1	char	-128 to 127
unsigned __int8	1	unsigned char	0 to 255
__int16	2	short, short int, signed short int	-32,768 to 32,767
unsigned __int16	2	unsigned short, unsigned short int	0 to 65,535
__int32	4	signed, signed int, int	-2,147,483,648 to 2,147,483,647
unsigned __int32	4	unsigned, unsigned int	0 to 4,294,967,295
__int64	8	long long, signed long long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned __int64	8	unsigned long long	0 to 18,446,744,073,709,551,615
Bool	1	none	false or true
Char	1	none	-128 to 127 by default 0 to 255 when compiled by using /J
signed char	1	none	-128 to 127
unsigned char	1	none	0 to 255
Short	2	short int, signed short int	-32,768 to 32,767
unsigned short	2	unsigned short int	0 to 65,535
Long	4	long int, signed long int	-2,147,483,648 to 2,147,483,647
unsigned long	4	unsigned long int	0 to 4,294,967,295
long long	8	none (but equivalent to __int64)	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long	8	none (but equivalent to unsigned __int64)	0 to 18,446,744,073,709,551,615

Enum	varies	none	
Float	4	none	3.4E +/- 38 (7 digits)
Double	8	none	1.7E +/- 308 (15 digits)
long double	same as double	none	Same as double
wchar_t	2	__wchar_t	0 to 65,535

Variables and Literals

Variables represent storage locations in the computer's memory. Literals are constant values that are assigned to variables.

Variables allow you to store and work with data in the computer's memory. They provide an "interface" to RAM. Part of the job of programming is to determine how many variables a program will need and what types of information they will hold.

Declaration of Variables:

Declaration of variables means that compiler assigns a memory location for the variable. Declaration of a variable consists of data type and the variable name.

Data type var_name;

e.g.

int number;

This is called a variable definition. It tells the compiler the variable's name and the type of data it will hold. This line indicates the variable's name is number.

int var, number;

float var, num, sum;

char ch;

Initialization of variables:

Initialization of variables assigns an initial value to the variable declared. Variable value is initialized by giving variable a name consisting of equal sign followed by a constant value.

Data type var_name = value;

e.g: int num=5;

float num=4.5, sum=0;

char ch='A';

Example 01:

```
// This program has a variable.
#include <iostream>
using namespace std;
int main()
{
        int number;
        number = 5;
        cout << "The value in number is " << number << endl;
        return 0;
}
```

Output:

The value in number is 5

Example 02:

```
// this program has variables of several types of the integer types.
 #include <iostream>
 using namespace std;
 int main ()
 {
        int checking;
        unsigned int miles;
        long days;

        checking = -20;
        miles = 4276;
        days = 189000;
        cout << "We have made a long journey of " << miles;
        cout << " miles.\n";
        cout << "Our checking account balance is " << checking;
        cout << "\nAbout " << days << " days ago Columbus ";
        cout << "stood on this spot.\n";
        return 0;
 }
```

Program Output:

We have made a long journey of 4276 miles.
Our checking account balance is -20
About 189000 days ago Columbus stood on this spot.

OPERATORS

Operators are in C++ to perform operations on variables and constants. Operators in C++ are:

Arithmetic Operators

Basic arithmetic operators in C++ are: *, /, -, +, ++, -- are the arithmetic operators.

Example

Suppose X and Y are two variables with X=5, Y=10

OPERATORS	DESCRIPTION	EXAMPLE
+	Adds both operands	X + Y = 15
-	Subtracts both operands	Y - X = 5
*	Multiply both operands	X * Y = 150
/	Divide both operands	Y / X = 2
%	Remainder after division of numbers	Y / X = 0

Assignment Operators:

Assignment operators are used to assign values to variables.

In the example below, we use the assignment operator (=) to assign the value 10 to a variable called x:

Example:

int x = 10;

Relational Operators

There are following relational operators supported by C++ language

Assume variable A holds 10 and variable B holds 20, then

Operator	Description	Example
==	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.

<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

Logical Operators:

There are following logical operators supported by C++ language.

Assume variable A holds 1 and variable B holds 0, then see examples

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false.	!(A && B) is true.

Compound Operators

They modify the variable by performing an operation on it.

Example:

```
#include <iostream>
using namespace std;
void main(){
        int a=0,b=1,c=2,d=4,e=8;
        a+=2;
        cout<<"Value of a is "<<a;
        b-=8;
        cout<<"Value of b is "<<b;
        c*=4;
        cout<<"Value of c is "<<c;
        d/=2;
        cout<<"Value of d is "<<d;
        e%=4;
```

```
	cout<<"Value of e is "<<e;
}
```

Output:
Value of a is 2
Value of b is -7
Value of c is 8
Value of d is 2
Value of e is 0

Input:

cin>>

The cin object in C++ is an object of class istream. It is used to accept the input from the standard input device i.e. keyboard. The cin corresponds to a standard input stream. The symbol “>>” refers to an *extraction* operator

Examples:

```
#include <iostream>

using namespace std;

int main()
{
	int x;
	/* For single input */
	cout << "Enter a number: ";
	cin >> x;
	cout << "X = " << x;
	return 0;
}
#include <iostream>
#include<string>
using namespace std;
void main()
{

	string exampleString;

	cout << "Enter string: ";

	cin >> exampleString; // assign input to exampleString

	cout << "You entered: " << exampleString << endl; // output exampleString

}
```

Lab Tasks:

Task 1

Suppose there are 7.48 gallons in cubic feet. Write a program that ask user to enter the number of gallons and then display the equivalent in cubic feet.

Task 2

Write a program that gets 2 integers input from user and store them in variables. Do the five basic Arithmetic Operations (+, - , *, /, %) of the two numbers. Print the results of operations as below.

```
Enter two integer numbers: 10 25
10 + 25 = 35
10 - 25 = -15
10 * 25 = 250
10 / 25 = 0.4
10 % 25 = 10
```

Task 3

Write a program that prompt user to input course name, obtained marks and total marks. Calculate the percentage using the below formula

marks_percentage = (marks_obtained / total)* 100

and display the results as follows.

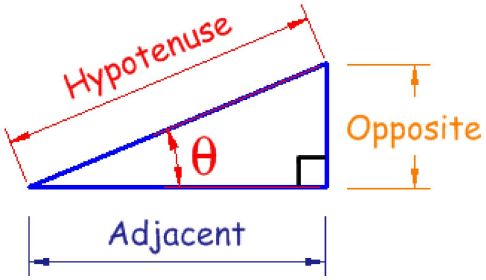
```
Enter Course name : Fundaments_Programming
Obtained Marks : 93
Total Marks : 100

In Fundaments_Programming Course. You have secured %93
```

Task 4

The distance between two cities (in km.) is input through the keyboard. Write a program to convert and print this distance in meters, feet, inches and centimeters.

Task 5

Hypotenuse refers to the side opposite the right angle in a right-angled triangle (as shown in the diagram below).



Area of this right-angled triangle can also be calculated using the following formula:

a = (½) * x * y

Note that:

h= hypotenuse

x= adjacent

Write a C++ program that prompt user to enter value of X and Y. You have to calculate the value of Area (a).

Task 6

Write a program that prompts the user to enter the weight of a person in kilograms and outputs the equivalent weight in pounds. Output both the weights rounded to two decimal places.