Institute of Computer Sciences & Information Technology (ICS&IT)
Faculty of Management and Computer Science (FMCS)
The University of Agriculture, Peshawar

Lab No. 3: Variables, Constants and Data Types

Objective:

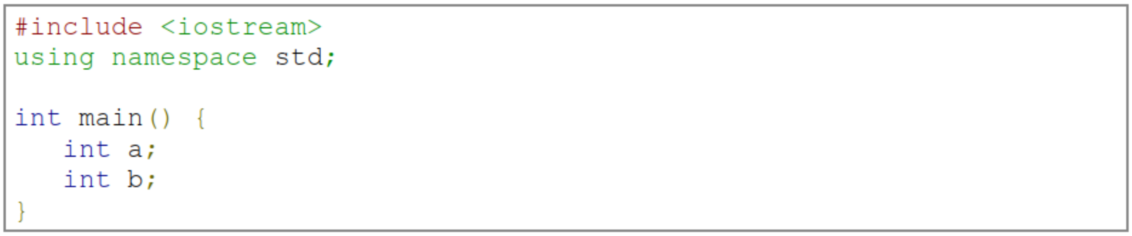
This lab is about familiarization with different data types, developing understanding of variables and constants taking input from user, showing output to screen and writing simple programs.

Variables:

Variables are the names you give to computer memory locations which are used to store information to be referenced and manipulated in a computer program.

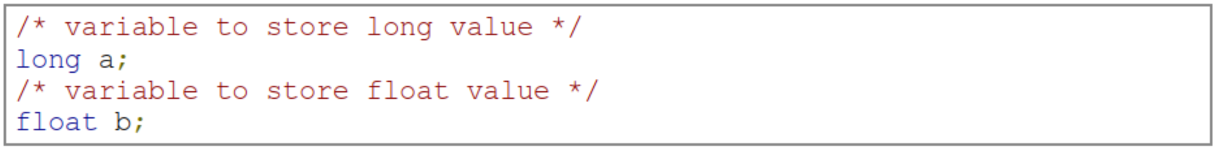
Creating variables:

Creating variables is also called declaring variables in C++ programming.



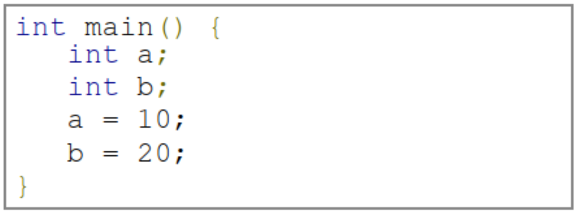
```
#include <iostream>
using namespace std;

int main() {
   int a;
   int b;
}
```

The above program creates two variables to reserve two memory locations with names a and b using int keyword to specify variable data type which means we want to store integer values in these two variables. Similarly, you can create variables to store long, float, char or any other data type. For example –

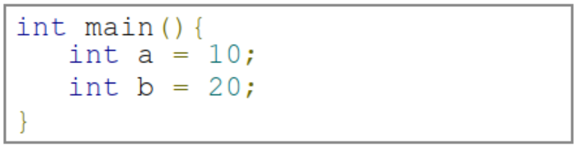


```
/* variable to store long value */
long a;
/* variable to store float value */
float b;
```

Store Values in Variables:



```
int main() {
   int a;
   int b;
   a = 10;
   b = 20;
}
```



```
int main(){
   int a = 10;
   int b = 20;
}
```

Get input from user:

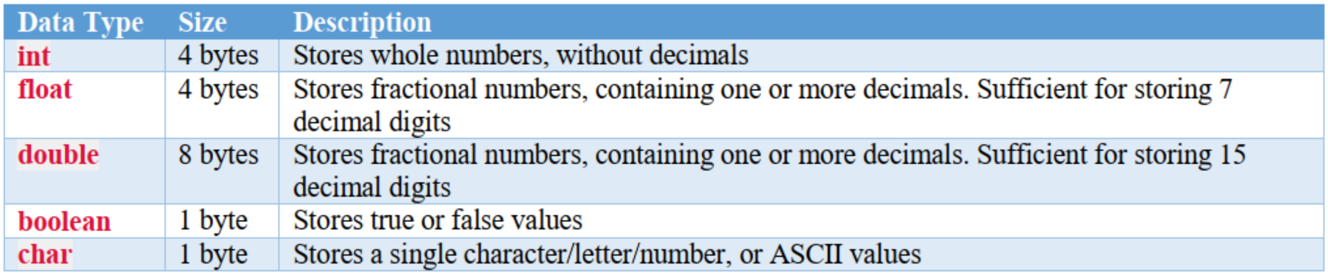
The cin object in C++ is an object of class istream. It is used to accept the input from the standard input device, i.e. keyboard. It is associated with the standard C input stream stdin.

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

int main() {
   int a, b, sum;
   cout<<”Enter first number”<<endl;
   cin>>a;
   cout<<”Enter second number”<<endl;
   cin>>b;
   sum = a+b;
   cout<<”Addition of two number is:”<<sum<<endl;

}
```

C++ Datatypes:

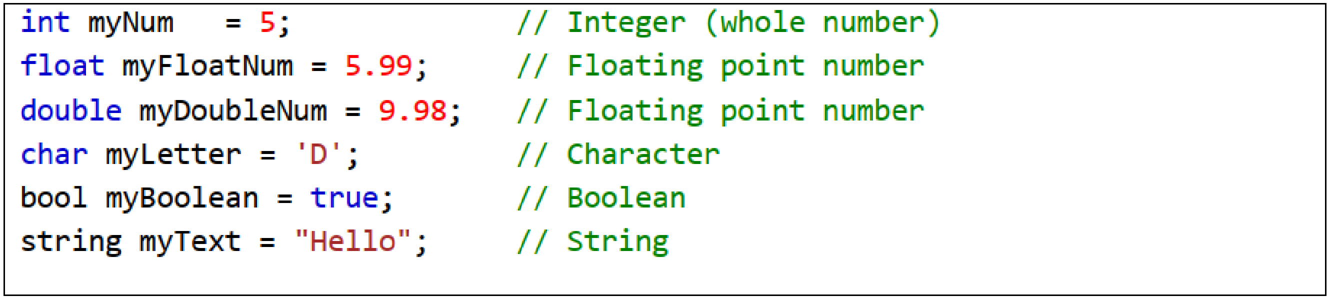
Every expression has a type – a formal description of what kind of data its value is. For instance, 0 is an integer, 3.142 is a floating-point (decimal) number, and "Hello, world!\n" is a string value(a sequence of characters). Data of different types take a different amounts of memory to store. Here are the built-in data types we will use most often:

A variable in C++ must be a specified data type. The data type specifies the size and type of information the variable will store:



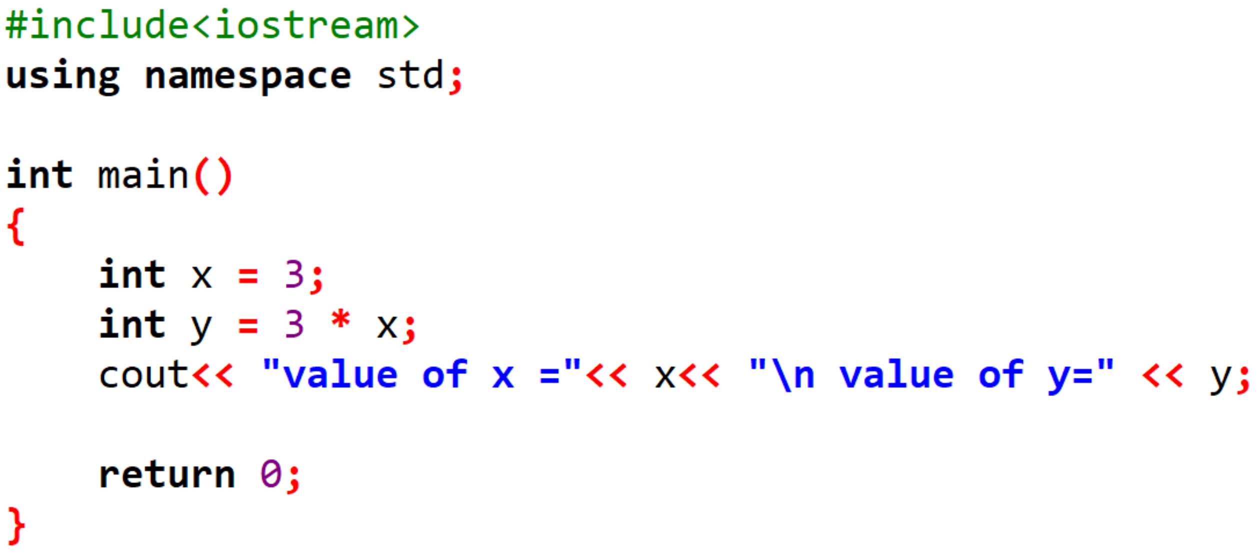
Data Type	Size	Description
int	4 bytes	Stores whole numbers, without decimals
float	4 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 7 decimal digits
double	8 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 15 decimal digits
boolean	1 byte	Stores true or false values
char	1 byte	Stores a single character/letter/number, or ASCII values

Example:



```
int myNum   = 5;             // Integer (whole number)
float myFloatNum = 5.99;     // Floating point number
double myDoubleNum = 9.98;   // Floating point number
char myLetter = 'D';         // Character
bool myBoolean = true;       // Boolean
string myText = "Hello";     // String
```

For Example:



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

int main()
{
    int x = 3;
    int y = 3 * x;
    cout<< "value of x ="<< x<< "\n value of y=" << y;

    return 0;
}
```

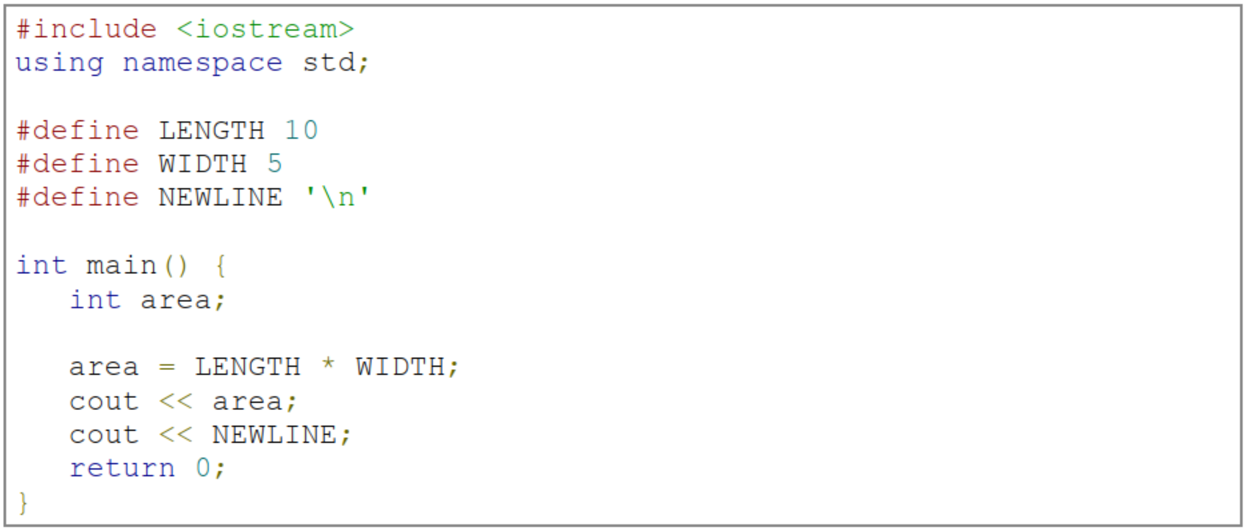
Defining Constants:

There are two simple ways in C++ to define constants –
Using **#define** preprocessor
Using **const** keyword

The #define Preprocessor:

Following is the form to use #define preprocessor to define a constant –

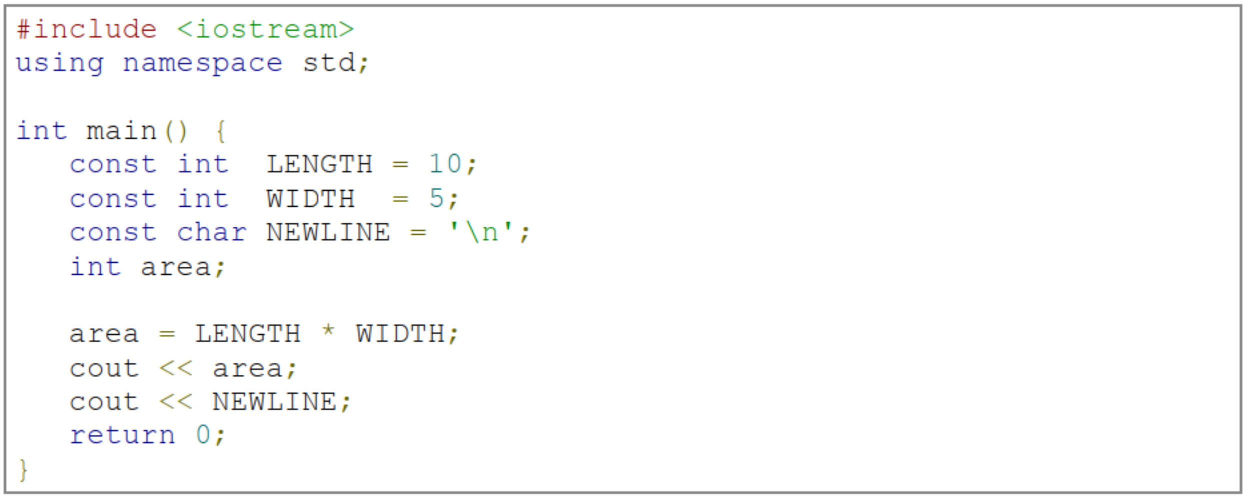
#define identifier value



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

#define LENGTH 10
#define WIDTH 5
#define NEWLINE '\n'

int main() {
   int area;

   area = LENGTH * WIDTH;
   cout << area;
   cout << NEWLINE;
   return 0;
}
```

The const Keyword:

You can use const prefix to declare constants with a specific type as follows - const type variable = value;



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

int main() {
   const int  LENGTH = 10;
   const int  WIDTH  = 5;
   const char NEWLINE = '\n';
   int area;

   area = LENGTH * WIDTH;
   cout << area;
   cout << NEWLINE;
   return 0;
}
```

Tokens:

Tokens are the minimal chunk of program that have meaning to the compiler –the smallest meaningful symbols in the language. Our code displays all 6 kinds of tokens, though the usual use of operators is not present here:

Token type	Description/Purpose	Examples
Keywords	Words with special meaning to the compiler	int, double, for, if
Identifiers	Names of sources in which input or any data is stored	cout, x1, cin
Literals	Basic constant values	24.3, “Hello world”
Operators	To perform operations like logical, arithematic etc	&, ǁ, +, -
Punctuation/Separators	Defining the structure of program	{,},(, ;
Whitespace	Spaces of various types use for formatting	Newline, tab, backslash

TASKS:

1. Write a C++ code that takes two numbers and displays the addition, subtraction, division, multiplication, and square of given numbers, on the console window.

2. Calculate the average age of a class of five students. Prompt the user to enter the age of each student.

3. Write a code in C++ that takes radius of a circle as input from user and outputs the circumference and area. The output should be clear and readable. Add proper comments to the code. You can set the value of π up to 3 decimal places.

4. Write a C++ program that takes temperature in Fahrenheit from user and convert it to Celsius and Kelvin using given formula.

C = (F – 32) / 1.8

K = C + 273

5. Write a program to calculate simple interest for a given P=4000, T=2, R=5.5.

Formula for simple interest: I = P*T*R/100

6. Write a C++ program to find the size of int, short int, float, double, bool and char. To find the size of variable, ***sizeof*** operator is used.