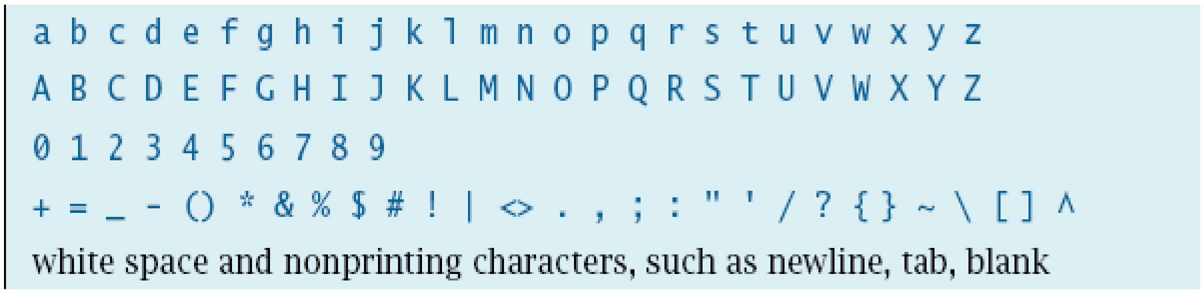
Introduction

We are going to study programming in C++ language , As you may know, C++ was built upon the foundation of C.

In fact, C++ includes the entire C language, and all C programs are also C++ programs. When C++ was invented, the C language was used as the starting point. To C++ were added several new features and extensions designed to support object-oriented programming (OOP). C was invented and first implemented by Dennis Ritchie . C++, invented at Bell Labs by Bjarne Stroustrup in the mid-1980s, is a powerful, modern, successor language to C. C++ adds to C the concept of *class*, a mechanism for providing user-defined types, also called *abstract data types*. C++ supports *object-oriented* programming by these means and by providing inheritance.

C++ Character set :

A program is composed of elements called *tokens*, which are collections of characters that form the basic vocabulary the compiler recognizes. The Following table shows the C++ character set.



```
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
+ = _ - () * & % $ # ! | <> . , ; : " ' / ? {} ~ \ [] ^
white space and nonprinting characters, such as newline, tab, blank
```

Keywords:

Keywords in C++ are explicitly reserved words that have a strict meaning and may not be used in any other way. They include words used for type declarations, such as int, char, and float; words used for statement syntax, such as do, for, and if; and words used for access control, such as public, protected, and private. The following table shows the keywords in C++ language.

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union
const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

Comments :

C++ has a single-line comment, written as // rest of line.

Example :

// My first program

A multiline comment is written as /* possibly multiline comment */ .

Everything Between /* and ***/** is a comment.

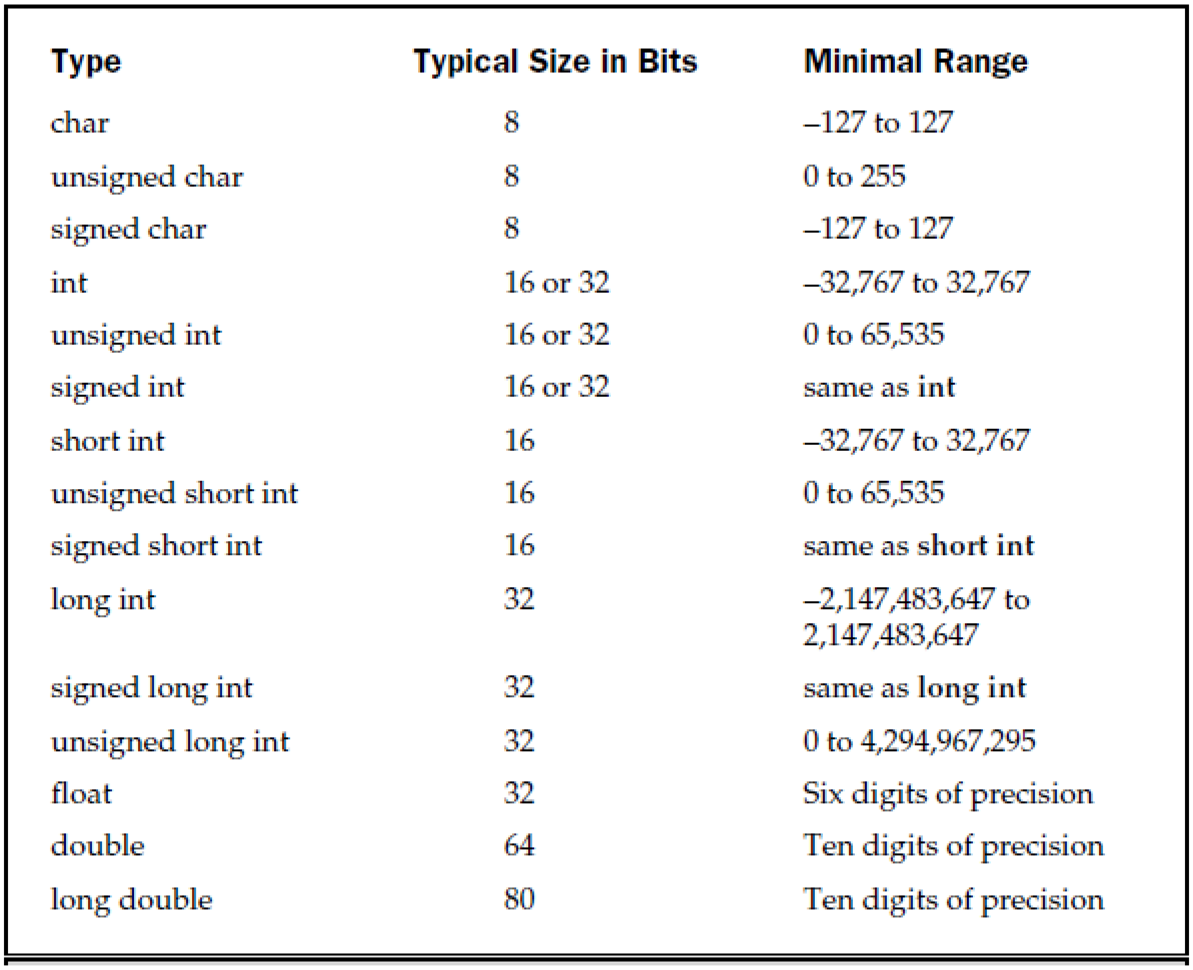
Example :

/ This name of Subject is Computer Science*

*This Lecture is Lecture Number four */*

The Five Basic Data Types :

There are five atomic data types in C++ : character, integer, floating-point, double floating-point, and valueless (**char**, **int**, **float**, **double**, and **void** respectively). As you will see, all other data types in C++ are based upon one of these types. Table below shows all valid data type combinations, along with their minimal ranges and approximate bit widths.



Type	Typical Size in Bits	Minimal Range
char	8	–127 to 127
unsigned char	8	0 to 255
signed char	8	–127 to 127
int	16 or 32	–32,767 to 32,767
unsigned int	16 or 32	0 to 65,535
signed int	16 or 32	same as int
short int	16	–32,767 to 32,767
unsigned short int	16	0 to 65,535
signed short int	16	same as short int
long int	32	–2,147,483,647 to 2,147,483,647
signed long int	32	same as long int
unsigned long int	32	0 to 4,294,967,295
float	32	Six digits of precision
double	64	Ten digits of precision
long double	80	Ten digits of precision

Identifier Names:

An identifier name (variables, functions, labels, and various other user-defined objects) in C++ is a sequence of letters, digits, and underscores.

An identifier cannot begin with a digit. Uppercase and lowercase letters are treated as distinct. *It is good practice to choose meaningful names as identifiers*. Here are some correct and incorrect identifier names:

Correct Identifier Name	Incorrect Identifier Name
x	8z
A	A@
count	1count
test23	hi!there
high_balance	high...balance

Variables:

As you probably know, a *variable* is a named location in memory that is used to hold a value that may be modified by the program. All variables must be declared before they can be used. The general form of a declaration is

type variable_list ;

Here, *type* must be a valid data type plus any modifiers, and *variable_list* may consist of one or more identifier names separated by commas. Here are some declarations:

char c ;

int i , j , R ;

double balance , profit , loss ;

Remember, in C/C++ the name of a variable has nothing to do with its type.

Arithmetic and Assignment Operators :

+	**Addition**	**c = a+b**
-	**Subtraction**	**c =a -b**
*	**Multiplication**	**c =a*b**
/	**Division**	**c =a / b**
%	**Modula (Remainder)**	**c =a % b**
=	**Assignment Operator**	**z = 8**

Arithmetic operators as per precedence (from high to low):

1	( )	for grouping the variables.
2	* / %	multiplication , division and Modula.
3	+ -	addition and subtraction.

The Mathematical Functions :

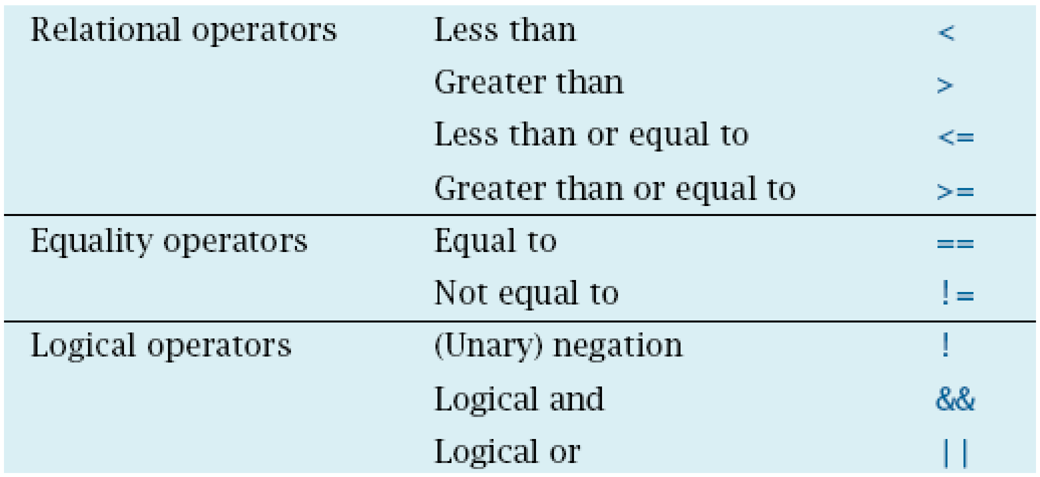
The **math.h** library contains the common mathematical function such as the following functions : **sin(x) , cos (x) , abs (x) , sqrt (x) , Pow (x,y), …**

Relational, Equality, and Logical Operators :

Just as with other operators, the relational, equality, and logical operators have rules of precedence and associativity that determine precisely how expressions involving them are evaluated. C++ systems use the bool values true and false to direct the flow of control in the various statement types.

The negation operator ! is unary. All of the other relational, equality, and logical operators are binary, operate on expressions, and yield the bool value, either false or true. Where a boolean value is expected, an arithmetic expression is automatically converted, in each case converting zero to false and nonzero to true.

The following table contains the C++ operators that are most often used to affect flow of control.



Relational operators	Less than	<
	Greater than	>
	Less than or equal to	<=
	Greater than or equal to	>=
Equality operators	Equal to	==
	Not equal to	!=
Logical operators	(Unary) negation	!
	Logical and	&&
	Logical or	||

Relational, Equality, and Logical operators per precedence (from high to low):

1	!
2	< , <= , > , >=
3	== , !=
4	&&
5	||

Input / Output Functions:

C++ input/output is not directly part of the language but rather is added as a set of types and routines found in a standard library. The C++ standard I/O library is **iostream** or **iostream.h**

The iostream library overloads the two bit-shift operators.

>> // "get from" input stream

<< // "put to" output stream

This library also declares two standard streams :

cin // standard in It is used to read an object from a standard input device (keyboard):

cin>>var.1>>var.2>>...>>var.n ;

cout // standard out, display an object onto the screen:

cout<<var.1<<var2<<...<<var.n ;

Example 1 : Write a C++ program to print “Welcome” on the Screen .

```
#include<iostream.h>
using  namespace  std;
int main ( )
{
  // A program to print welcome
    cout << "Welcome";

     return 0 ;
}
```

Output :

```
Welcome
```

Example 2 : Write a C++ program to read (input) three integer numbers and find and print the Average of these numbers .

```
#include<iostream.h>
using  namespace  std;
 int  main ( )
  {
      int x,y,z ;
      double av ;
          cout << "Enter Three Numbers" << endl ;
          cin>>x>>y>>z ;
          av= (x+y+z) / 3.0  ;
          cout << "The average is: " << av ;

           return 0 ;
 }
```

WORK SHEET

Exercise 1. Write a C++ program to read the length and width of rectangle and find and print the area of this rectangle.

Exercise 2. Write a program to input three integer numbers from the keyboard and print the sum and product of these numbers .

Exercise 3. Write a program to input a double number as temperature and Convert a temperature from degrees Celsius (centigrade) to degrees Fahrenheit

Exercise 4. Write a program that will calculate the price for a quantity entered from the keyboard, given that the unit price is $5 .

Exercise 5. Write a program to input the quantity and unit price then calculate the price for a quantity .

Exercise 6. Find the value of **Z** for the following (Manually without writing program) :

Z = (6 + 8 * 4 + ((3 + 2 -4) * 7) + 3) / 2

Exercise 7. Write a C++ program to input integer number **x** and find and print the **sin(x) , cos (x) , sqrt (x)** using **math.h** library functions .