Unit 3 Introduction To c++

C++ Character Sets:

Letters	A-Z , a-z
Digits	0-9
Special Symbols Space	+ - * / ^ \ ( ) [ ] { } = != <> . „ “ $, ; : % ! & ? _ # <= >= @
White Spaces	Blank spaces, horizontal tab, carriage return
Other Characters	Any of the 256 ASCII character

Token:-The smallest individual unit in a program is known as token. Tokens used in C++ are:

KEYWORDS

Keywords are the certain reserved words that convey a special meaning to the compiler. These are reserve for special purpose and must not be used as identifier name.eg for , if, else , this , do, etc.

LITERALS

The data items which never change their value throughout the program run. There are several kinds of literals:

•Integer literals

• Character literals

•Floating literals

•String literals

Integer literals

Integer literals are whole numbers without any fractional part. An integer literal must have at least one digit and must not contain any decimal point. It may contain either + or - sign. A number with no sign is assumed as positive. C++ allows three types of integer literals:

(i) Decimal Integer Literals :- An integer literal without leading 0 (zero) is called decimal integer literals e.g., 123, 786 , +97 , etc.

(ii) Octal Integer Literals :- A sequence of octal digit starting with 0 (zero) is taken to be an octal integer literal (zero followed by octal digits). e.g., 0345, 0123 , etc.

(iii) Hexadecimal Integer Literals :- Hexadecimal Integer Literals starts with 0x or 0X followed by any hexa digits. e.g., 0x9A45, 0X1234, etc.

Character literals

Any single character enclosed within single quotes is a character literal. e.g ‘ A’ , ‘3’

Floating literals

Numbers which are having the fractional part are referred as floating literals or real literals. It may be a positive or negative number. A number with no sign is assumed to be a positive number. e.g 2.0, 17.5, -0.00256

String Literals

It is a sequence of character surrounded by double quotes. e.g., “abc” , “23”. PUNCTUATORS: The following characters are used as punctuators which are also known as separators in C++

[ ] { } ( ) , ; : * ..... = #

Punctuator	Name	Function
[ ]	Brackets	These indicates single and multidimensional array subscripts
( )	Parenthesis	These indicates single and multidimensional array subscripts
{ }	Braces	Indicate the start and end of compound statements.
;	Semicolon	This is a statement terminator.
,	Comma	It is used as a separator.
:	Colon	It indicates a labeled statement
*	Astrisk	It is used as a pointer declaration
..	Ellipsis	These are used in the formal argument lists of function prototype to
=	Equal to	It is used as an assigning operator.
#	Pound singn...	This is used as preprocessor directives.

OPERATORS

An operator is a symbol or character or word which trigger some operation (computation) on its operands.

(i) Unary operators :- Those which require only one operand to operate upon. e.g. unary - , unary + , ++ , - -

(ii) Binary operators :- Binary operators require two operands to operate upon. e.g. +, *, /, -, etc.

(iii) Ternary Operator :- Ternary operator require three operands to operate upon. Conditional operator (? :) is a ternary operator in C++.