

# SUBJECT : COMPUTER SCIENCE

## UNIT 1 : PROGRAMMING IN C++

### Introduction to C++

- C++ programming language developed by AT&T Bell Laboratories in 1979 by Bjarne Stroustrup. C++ is fully based on **Object Oriented Technology** i.e. C++ is ultimate paradigm for the modeling of information.
- C++ is the successor of C language.
- It is a case sensitive language.

**Character Set-** Set of characters which are recognized by c++compiler i.e Digits 0 -9), Alphabets A -Z & a-z and special characters + - \* , . “ ‘ < > = { ] space etc **i.e 256 ASCII characters.**

**Tokens-** Smallest individual unit. Following are the tokens

- **Keyword-** Reserve word having special meaning the language and can't be used as identifier.
- **Identifiers-** Names given to any variable, function, class, union etc. Naming convention rule for writing identifier is as under:
  - i First letter of identifier is always alphabet.
  - ii Reserve word cannot be taken as identifier name.
  - iii No special character in the name of identifier except under score sign ‘\_’.
- **Literals-** Value of specific data type assign to a variable or constant. Four type of Literals:
  - i Integer Literal i.e **int x =10**
  - ii Floating point Literal i.e **float x=123.45**
  - iii Character Literal i.e **char x= ‘a’**, **enclosed in single quotes and single character only.**
  - iv String Literal i.e cout<< **“Welcome”** , **anything enclosed in double quotes**
- **Operator** – performs some action on data
  - Arithmetic+, -, \*, /, %
  - Assignment operator (=
  - Increment / Decrement (++ , --
  - Relational/comparison (<, >, <=, >=, ==, !=.
  - LogicalAND&&, OR ||), NOT!
  - Conditional (? :

### Precedence of operators:

++post increment, --post decrement Highest

Low

++pre increment, --pre decrement, sizeof !not, -unary, +unary plus)

\*(multiply, / (divide, % modulus)

+add), -subtract

<less than), <=less than or equal, >greater than), >=greater than or equal to)

==equal, !=not equal

&& (logical AND

||logical OR

?:conditional expression

=simple assignment and other assignment operators(arithmetic assignment operator

, Comma operator

• **Punctuation** – used as separators in c++ e.g. [ { } ] , ; # = : etc

**Data type-** A specifier to create memory block of some specific size and type. C++ offers two types of data types:

**1 Fundamental type :** Which are not composed any other data type i.e. int, char, float and void

**2 Derived data type :** Which are made up of fundamental data type i.e array, function, class, union etc

7

**Data type conversion-** Conversion of one data type into another data type. Two type of conversion i.e

i Implicit Conversion – It is automatically taken care by compiler in the case of lower range to higher range e.g. **int x, char c='A' then x=c** is valid i.e character value in c is automatically converted to integer.

ii Explicit Conversion- It is user-defined that forces an expression to be of specific type. e.g. **double x1,x2 and int res then res=intx1+x2)**

**Variable-** Memory block of certain size where value can be stored and changed during program execution. e.g. **int x, float y, float amount, char c;**

**Constant-** Memory block where value can be stored once but can't changed later on during program execution.e.g. **const int pi =3.14;**

**cout** – It is an object of **ostream\_withassign** class defined in iostream.h header file and used to display value on monitor. Eg. **Cout<< “string”<< variable;**

**cin** – It is an object of **istream\_withassign** class defined in iostream.h header file and used to read value from keyboard for specific variable. Eg **cin>> variable**

**comment-** Used for better understanding of program statements and escaped by the compiler to compile . e.g. – **single line (// and multi - line/\*....\*/**

**Cascading** – Repeatedly use of input or output operators( “>>” or “<<<” in one statement with cin or cout.

**Control structure:**

**Sequence control**

**if else**

**Multiple Choice Statement**

**If –else-if**

**Switch Statement**

**Alternate for if - else- if works for only exact match**

**loop control statement**

**while ,do... while, for**  
Syntax Syntax Syntax Syntax Syntax  
ifexpression  
n)  
{  
statements;  
}  
Ifexpression  
n)  
{  
statements;  
}  
else  
{  
statements;  
}  
If (expression)  
{  
statements  
}  
else  
ifexpression)

```

{
statement
}
else
{
statement
}
switchint / char
variable
{ case literal1:
[statements
break;}
case literal2:
[statements,
break;}
default:statements;
}

```

**Break** is compulsory statement with every **case** because if it is not included then the controls executes next case statement until next break encountered or end of **switch** reached.

**Default** is optional, it gets executed when no match is found

```

whileexpression
{
statements;
}

```

**Entry control loop works for true condition.**

```

do
{
statements;
} whileexpression;

```

**Exit Control Loop execute at least once if the condition is false at beginning.**

**for loop**

```
for expression1;expressio  
n2;expression3)
```

```
{  
statement;  
}
```