

**INTRODUCTION TO C++**

- **C++ CHARACTER SET:** Character set is a set of valid characters that a language can recognize. A character represents any letter, digit or any other special character. The C++ programming language also has some character set.

Types	Character Set
Uppercase Alphabets	A, B, C, ... Y, Z
Lowercase Alphabets	a, b, c, ... y, z
Digits	0, 1, 2, 3, ... 9
Special Symbols	~ ' ! @ # % ^ & * ( ) _ - + =   \ { } [ ] ; ; " ' < > , . ? /
White spaces	Single space, tab, new line.

- **BASIC DATA TYPES:** C++ supports a large number of data types. The built in or basic data types supported by C++ are integer, floating point and character type.
- **TOKENS:** A token is a group of characters that logically belong together. The programmer can write a program by using tokens. C++ uses the following types of tokens.
  - Keywords
  - Identifiers
  - Literals
  - Punctuators
  - Operators
- **KEYWORDS:** There are some reserved words in C++ which have predefined meaning to compiler called keywords. Some commonly used keywords are given below:

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

- **IDENTIFIERS:** The identifier is a sequence of characters taken from C++ character set. The rules for the formation of an identifier are:
  - (i) An identifier can consist of alphabets, digits and/or underscores.
  - (ii) It must not start with a digit.
  - (iii) C++ is case sensitive, i.e., upper case and lower case letters are considered different from each other. It may be noted that TOTAL and total are two different identifier names.
  - (iv) It should not be a reserved word (keywords).
- **LITERALS:** Literals (often referred to as constants) are data items that never change their value during the execution of the program. The types of literals available in C++ are integer-constants, character-constants, floating-constants and string-literals.
- **PUNCTUATORS:** The following characters are used as punctuators in C++ : Brackets [ ] , Parentheses ( ) , Braces { } , Comma ,, Semicolon ; , Colon : , Asterisk \* , Equal sign = , Pound sign # .

- **OPERATORS:** Operators are special symbols used for specific purposes. C++ includes many operators such as Arithmetical operators, Relational operators, Logical operators, Unary operators, Assignment operators, Conditional operators and Comma operator.
- **THE SIZEOF OPERATOR:** The size of operator determines the amount of memory required for an object at compile time rather than at run time.
- **THE ORDER OF PRECEDENCE:** The order in which the operators are used in a given expression is called the order of precedence. The following table shows the order of precedence:

( ) [ ]	Operators within parenthesis are performed first	Higher	
++, --	Postfix increment / decrement		
+, -	Prefix increment / decrement		
*, /, %	Multiplication, Division, Modulus		
+, -	Addition, Subtraction		
<, <=, >, >=	Less than, Less than or equal to, Greater than, Greater than or equal to		
==, !=	Equal to, Not equal to		
&&	Logical AND		
	Logical OR		
?:	Conditional Operator		
=	Simple Assignment		
+=, -=, *=, /=	Shorthand operators		
,	Comma operator		Lower

- **TYPE CONVERSION:** The process in which one pre-defined type of expression is converted into another type is called conversion. There are two types of conversion in C++.
  - Implicit conversion : When two operands of different types are encountered in the same expression, the lower type variable is converted to the higher type variable
  - Explicit conversion: It is also called type casting. It temporarily changes a variable data type from its declared data type to a new one. It may be noted here that

type casting can only be done on the right hand side of the assignment statement.

- **CONSTANTS:** A number which does not change its value during execution of a program is known as a constant. Any attempt to change the value of a constant will result in an error message. A constant in C++ can be of any of the basic data types.
- **VARIABLES:** It is a location in the computer memory which can store data and is given a symbolic name for easy reference. The variables can be used to hold different values at different times during the execution of a program. Examples of valid variable declarations are: `int count; int i, j, k; char ch, first; float total, Net; long int sal;`
- **INPUT / OUTPUT (I/O):** C ++ supports input/output statements which can be used to feed new data into the computer or obtain output on an output device such as: VDU, printer etc. It provides both formatted and unformatted stream I/O statements. In addition to the above I/O streams, two operators << and >> are also used. The operator << is known as put to or bit wise shift operator. The operator >> is known as extraction or get from operator.
- **STRUCTURE OF C++ PROGRAM:** The structure of a C++ program is given below:

```
# include <header file>
main ( )
{
.....
.....
.....
}
```

- A C++ program starts with function called main( ). The body of the function is enclosed between curly braces. The program statements are written within the braces. Each statement must end by a semicolon (statement terminator). A C++ program may contain as many functions as required. However, when the program is loaded in the memory, the control is handed over to function main( ) and it is the first function to be executed.

### CHECK YOURSELF

1. Which of the following is the correct identifier?  
A. \$var\_name  
B. VAR\_123  
C. varname@  
D. None of the above
2. Which of the following is the correct syntax to print the message in C++ language?  
A. cout <<"Hello world!";  
B. Cout << Hello world! ;  
C. Out <<"Hello world!;  
D. None of the above
3. C++ is a \_\_\_ type of language.  
A. High-level Language  
B. Low-level language  
C. Middle-level language  
D. None of the above
4. Which of the following is called address operator?  
A. \*  
B. &  
C. \_  
D. %
5. Which of the following is called insertion/put to operator?

- A. <<
- B. >>
- C. >
- D. <

### STRETCH YOURSELF

1. Write a C++ program to print first 10 natural numbers?
2. Write a C++ program to print first 10 multiples of 2?
3. What is the use of sizeof operator? Explain with example?
4. Create any 5 identifiers as per the rules to be followed while naming an identifier.
5. Write a C++ program to read value of any three variables from user and print them.

### ANSWERS

Answers to Check Yourself:

1. B
2. A
3. C
4. B
5. A