Introduction

Computer Languages:

a. Low-Level Languages:
   a. Machine-level language
   b. Assembly language
b. High-Level Language: that resembles the human language: FORTRAN, COBOL, BASIC, PL/I, Pascal, C, C++…
   - The *FORTRAN* programming language was conceived in the early 1950s the name produced from the two words FORmula TRANslation.
   - **COBOL** (an acronym for Common Business-Oriented Language) is a compiled English-like computer programming language designed for business use. It is imperative, procedural and, since 2002, object-oriented. **COBOL** is primarily used in business, finance, and administrative systems for companies and governments
   - *BASIC* (an acronym for Beginner's All-purpose Symbolic Instruction Code) is a family of general-purpose, high-level *programming* languages whose design philosophy emphasizes ease of use. In 1964, John G. Kemeny and Thomas E. Kurtz designed the original *BASIC language* at Dartmouth College in New Hampshire.
   - Programming language 1 (PL/1) was born because IBM wanted to design a machine that would supersede all the IBM architectures that came before it to become the common machine architecture for the business and scientific communities. This became the IBM System 360.

Translators:

Because machine understand only the machine language, therefore it is imperative to translate the all languages above machine-language into machine language i.e. the language of 1’s and 0’s. Different translators are:

Assemblers
Compilers
Interpreters

Types of Softwares/Programs

a. System Software
b. Application Software
   i. General Purpose: Wordstar, word, excel, DbaseIII,
   ii. Specific Purpose: Payroll, inventory control, Library management

iii. Scientific Application

iv. Business Application

Difference between C and C++

	C	C++
Inventors	C was developed by Dennis Ritchie between 1969 and 1973 at AT&T Bell Labs	C++ was developed by Bjarne Stroustrup in 1979 with C++'s predecessor "C with Classes"
Type of language	It is procedural language	It is object oriented language
Calling	C is function-driven; i.e. Functions are the building blocks of a C program	C++ is object-driven; i.e. objects are building blocks of a C++ program.
Set	It is a subset of C++	Superset of C
Comments	/*… */	// for single line /* */ for multi-line
Declaration of data members	Are to be declared at the beginning of the program	Can be declared anywhere in the program but before its actual usage
Mapping between Data and Function	It is difficult and complicated in C	Mapping between Data and Function can be used using "Objects"
Stream handling	Through scanf() and printf() functions	Through cout and cin objects of class istream and ostream
Global Declaration	Multiple Declaration of global variables are allowed	Multiple Declaration of global varioables are not allowed

Structure	struct strruc_name{
	//declarations
	};
	struct strruc_name var1,var;
		struct strruc_name{
		//declarations
		};
		strruc_name var1,var;

	structures cannot contain functions in C	functions can be used inside a structure

Enumeration	enum Day {Mon, Tue, Wed, Thu,Fri, Sat, Sun};
	/* then we create an enum variable */
	enum Day day; /*then we can use */
	day=4;
		Enum Day {Mon, Tue, Wed, Thu,Fri, Sat, Sun};
		// then we create an enum var
		Day day; //then we can use
		day=Fri;

Type casting	(int *) expression	int * (expression) we could use type def for data type as:

		typedef int* roll; then use roll(expression)
Memory allocation and deallocation	malloc() and calloc() Functions are used for Memory Allocation and free() function for memory Deallocating.	new and delete operators are used for Memory Allocating and Deallocating.
Exception Handling	Exception Handling is not present.	Exception Handling is done with Try and Catch block
Function and Operator Overloading	Not possible in C	It is possible in C++ to overload an operator
Program Design Methodology	Top down approach is used in Program Design	Bottom up approach adopted in Program Design
namespace	No namespace Feature is present in C Language	Namespace Feature is present in C++ for avoiding Name collision
Data security	In case of C, the data is not secured. data are free entities and can be manipulated by outside code. This is because C does not support information hiding	the data is secured(hidden) in C++. Encapsulation hides the data to ensure that data structures and operators are used as intended
class	Do not exist; hence OOPS topics like objects, constructor, inheritance etc not supported	New concept of object and classes, hence OOPS topics like objects, constructor, inheritance etc are supported
template	C does not support the template	Templates are supported in C++

Structure of a C++ Program

C++ Programming language is most popular language after C Programming language. C++ is first Object oriented programming language. We have summarize structure of C++ Program in the following Picture –

I. Header File Declaration Section
II. Global Declaration Section
III. Class Declaration Section
IV. Main Function
V. Method Definition Section

Section 1 : Header File Declaration Section

1. Basically all preprocessor directives are written in this section
2. Header files are listed here. They provide <u>Prototype declaration</u> for different library functions.
3. We can also include **<u>user define header file</u>**.

Section 2 : Global Declaration Section

1. Global Variables are declared here. These may include –
    - Declaring Structure
    - Declaring Class
    - Declaring Variable

Section 3 : Class Declaration Section

Class declaration and all methods of that class are defined here.

Section 4 : Main Function

1. Each and every C++ program always starts with main function.
2. This is entry point for all the function. Each and every method is called indirectly through main.
3. Class Objects in the main.
4. Operating system calls this function automatically.

```
int main()

{

        --

        Return 0;

}
```

- Main function: on its left hand side it uses a return type. Main is expected to return a value to the operating system. A ‘0’ is returned as a status indicator that the program was a success and no error. Any value other than ‘0’ indicates an error. Every function (including the main function) definition specifies four elements: return type, function name, a (possible empty) parameter list enclosed in parentheses and the function body.
- Function body is a list of block of sequential statements which start with opening curly braces and ends with closing curly braces.

Section 5 : Method Definition Section

- This is optional section. Generally this method was used in C Programming.

Example of a C++ program

```
#include<iostream.h>
class abc
{
            private:
                        int x, y;
            public:
                        void input()
                        {
                                    cout<< "input the no\n";
                                    cin >>x>>y;
                        }
                        void disp();
};
void abc:: disp()
            {
                        cout <<"the numbers are :"<<endl<<x<<endl<<y;
            }
void main()
{
            abc z;
            z.input();
            z.disp();
 }
```

Compiling the Program

Compilers with IDE(e.g. Turbo CPP) have short cut keys/menu options for compiling and running the programs. They also provide command line interface for compiling and running. Others like g++ provide an exclusive command line interface for this purpose.

Procedure:

a. IDE based compilers:

   F9 for compilation

   CTRL + F9 for execution

b. Command line based compilers

   Like G++; the procedure for compilation and executing is as:

   Compilation:

```
$: g++ filename.cc –o filename
```

OR

```
$: cc filename.cc
```

This will create an exe filea.exe; if you want to name the executable the same as name of file or some other name, use the switch option –o followed by file name as:

```
cc filename.cc –o filename
```

This will generate filename as the exe program. The exe file can be executed by simply writing the filename on the command prompt and pressing the enter key.