**PARTS OF C++ PRORAM :**

**HEADER FILES –**

They are the storehouse for all library utilities packaged with C++

**MAIN FUNCTION-**

This is the place where all the instructions that a program should have are contained.

**TOKEN-**

It is the smallest identifiable part of C++ program.

**IDENTIFIER-**

All the symbols created by the programmer for his own use must be given a name, those names are called identifiers.

Conditions-

Only alphabets (both upper and lower case), digits and underscore can be used

Keywords can’t be used as identifiers

Names can’t start with a digit

**Keywords-**

Those words that have a specific meaning to the compiler are keywords

TYPE MODIFIER-

These are keywords used to modify the capacity of a primitive data type.

They include-

Short, Long, Signed, Unsigned

**VARIABLES-**

These are storage places where different values can be stored.

**STAGES OF C++ PROGRAMS-**

**SOURCE PROGRAM-**

The code that is written by the programmer is termed as source code program.

Its extension is .cpp

**OBJECT PROGRAM-**

Source program is sent to the compiler to check whether any error exists or not.

If yes, the source code is rejected and sent back for modification and steps are repeated.

If no, the compiler converts the source program into computer understandable form called as object program.

Its extension is .obj

**EXECUTABLE FILE PROGRAM-**

The object program is sent to various external utilities like linkers and loaders to create executable program.

It has extension .exe

**PROGRAMMING METHODOLOGY**

**Best features of a good program:**

a. **Modular approach**

The entire solution of a problem is broken down into smaller manageable fragments.

b. **Clarity and Simplicity of Expressions**

The more the expressions are simple to understand, the better they are readable to the unknown programmers.

c. **Use of proper Names for identifiers**

A suitable name given for identifiers depicts the purpose clearly to the layman programmers that adds to the understanding of the program.

d. **Comments**

Those lines of instructions that are not read by the compiler are termed as comments.

Types of comments:

a) Single line: Hides a single line of code

//int a 5;

b) Multiple line: Hides multiple lines of code

/\*int a=5;

Cout<<a;

\*/

e. **Indentation**

The process of writing the codes in fixed margin for better readability

f. **Documentation**

The process of giving proper comments or descriptions of the code in a program.

g. **Program Maintenance**

It refers to the process of the installation of the software and time to time updation as identified by the user.

h. **Debugged programs**

The process of removing errors from a program.

**Error**

Any portion of a program that is cannot be processed and executed is called as an error.

**Syntax Errors**

Those instructions that are not understood by the compiler.

a) Semi colon missing after completion of a statement

b) Prototype missing

**Run-Time Errors**

Those instructions that are identified as unexecutable by the computer processor at the time of program execution.

a) Division by zero

b) Square root of a negative number

**Logical Errors**

Those instructions that yield an unexpected output, i.e. when the actual output doesn’t match the desired output.

a) Use of wrong operator/formula

b) Use of wrong logic

**Problem Solving Methodologies**

* Understanding of the problem- The purpose of the program must first be learnt input and output requirement is identified
* Solution for the problem -A suitable methodology to be worked out after the objective of the program is understood
* Identifying minimum number of inputs required for output-The user of a program must be allowed to enter the least number of inputs as the more the number of inputs, more is the chances of making errors
* Writing code to optimizing execution time and memory storage-Usage of least number of variables without sacrificing the purpose of the program
* Step by step solution for the problem - A meaningful sequence to be maintained so as to produce the identified result
* Breaking down the solution into simple steps- Modular approach- Division of the code into independent and identifiable fragments, to ensure proper maintenance of the code
* Identification of Arithmetic and logical operations required for solution - A proper specification of identification of mathematical and logical operations must be generated
* Algorithm-A step by step procedure to solve a give problem
* Flowchart-A diagrammatical representation of an algorithm to solve a problem.
* Control Structure - Various control structure must be used to ensure a meaningful and
  + Conditional control - Use of conditional expressions covering all possibilities
  + Looping-
    - Finite-A loop that has a definite end
    - Infinite -A loop that never ends, as the condition specified remains true forever