BASICS OF C++ & PROGRAMMING

When you are studying computer science basics or even the advanced skills, first things first comes the concept of problem solving.

What is problem solving?

Since we are learning a technical subject, everything will be said and done in technical terms only.

In Computer science terms problem solving is a combination of different steps. The steps include

- **Analyzing the given problem.**
- **Break down problem into smaller sub parts to get a clear idea of what the problem really consists of.**
- **Write down the solution on paper**
- **Verify your solution on paper.**
- **Write code.**

Flowcharts

Flowcharts helps in achieving the solution to a given problem on a paper. It is a part of the problem solving.

Formal definition of flowchart is that it is a diagrammatic representation of a given problem.

Not only does It help in achieving the solution but it breaks down the process into parts and display them in a visually pleasing way.

Benefits of using a flowchart

- **There are less errors when you write a code with the help of flowchart.**
- **It helps in documentation.**

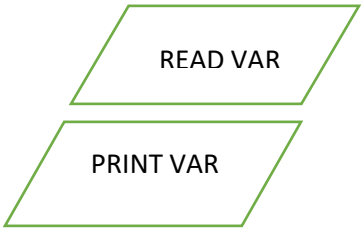
Flowcharts Components

- **The first component is the TERMINATOR. Diagrammatic representation of this is**

Mainly used to denote the start point of the program.

Used to denote the end point of the program.

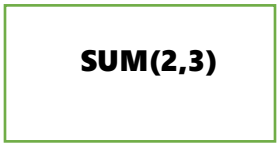
- **INPUT/ OUTPUT:**



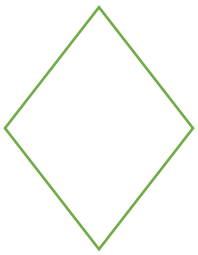
Used for taking input from the user and store it in variable 'var'.

Used to output value stored in variable 'var'.

- **PROCESS: Used to perform the operation(s) in the program. For example: Sum(2, 3) just performs arithmetic summation of the numbers 2 and 3.**

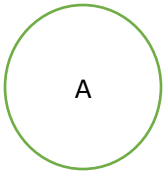


- **DECISION: Used to make decision(s) in the program means it depends on some condition and answers in the form of TRUE(for yes) and FALSE(for no). Diagrammatic representation is given below.**

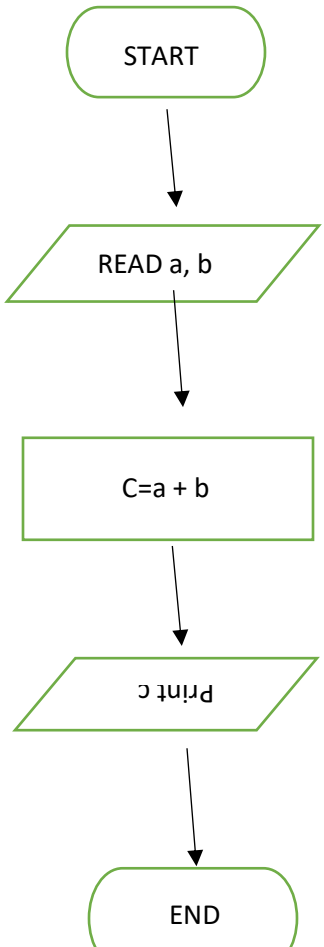


- **ARROWS: Generally, used to show the flow of the program from one step to another. The head of the arrow shows the next step and the tail shows the previous one.**

- **CONNECTORS: Used to connect different parts of the program and are used in case of break-through. Generally, used for functions(which we will study in our further sections).**



Example 1: Suppose we have to make a flowchart for adding 2 numbers a and b.



GETTING STARTED

C++ code begins with the inclusion of header files. There are many header files available in the C++ programming language which you will discuss while moving ahead with the course. So, what are these header files? The names of program elements such as variables, functions, classes, and so on must be declared before they can be used. For example, you can't just write x = 42 without first declaring variable 'x' as:

```
int x=15;
```

The declaration tells the compiler whether the element is an int, a double, a float, a function or a class. Similarly, header files allow us to put declarations in one location and then import them wherever we need them. This can save a lot of typing in multi-file programs. To declare a header file, we use #include directive in every .cpp file. This #include is used to ensure that they are not inserted multiple times into a single .cpp file.

Note: # operator is known as Macros.

Now, moving forward to the code

```
#include <iostream>

using namespace std;
```

iostream stands for Input/Output stream, meaning this header file is necessary if you want to take input through the user or print output to the screen. This header file contains the definitions for the functions:

- **cin : used to take input**

• cout : used to print output namespace defines which input/output form is to be used.

You will understand these better as you progress in the course. Note: semicolon (;) is used for terminating a C++ statement. ie. different statements in a C++ program are separated by semicolon.

Main function():

Look at the following piece of code:

```
int main() {
Statement 1;
Statement 2; ... }
```

You can see the highlighted portion above. Let's discuss each portion stepwise. Starting with the line:

```
Int main()
```

• int : This is the return-type of the function. You will get this thing clear once you reach the Functions topic.

• main() : This is the portion of any C++ code inside which all the commands are written and gets executed.

○ This is the line at which the program will begin executing. This statement is similar to the start block of flowcharts.

○ As you will move further in the course, you will get a clear glimpse of what this function is. Till then, just note that you will have to write all the programs inside this block

. • {} : all the code written inside the curly braces is said to be in one block, also known as scope of a particular function. Again, these things will be clear when you will study functions. For now, just understand that this is the format in which we are

gonna write our basic C++ code. From time to time as you will move forward with the course, you will get a clear and better understanding.

DECLARING A VARIABLE:

To declare a variable, we should always know what type of value it should hold, whether it's an integer (int), decimal number (float, double), character value (char). In general, the variable is declared as follows:

```
Datatype variableName = VALUE;
```

Note: Datatype is the type of variable:

- **int : Integer value**
- **float, double : Decimal number**
- **char : Character values (including special characters)**
- **bool : Boolean values (true or false)**
- **long : Contains integer values but with larger size**
- **short : Contains integer values but with smaller size Table for datatype and its size in C++: (This can vary from compiler to compiler and system to system depending on the version you are using)**

Datatype	Default size
Bool	1 byte
Char	1 byte
Short	2 bytes
int	4 bytes
long	8 bytes
float	4 bytes
double	8 bytes

For example: To declare an integer variable 'a' with a value of 5, the structure looks like:

```
int a = 5;
```

Similarly, this way other types of variables can also be declared. There is one more type of variable known as string variables which store combinations of characters. You will study that in your further noes.

Rules for variable names:

- **Can't begin with a number.**
- **Spaces and special characters except underscore(_) are not allowed.**
- **C++ keywords (reserved words) must not be used as a variable name.**
- **C++ is case-sensitive, meaning a variable with name 'A' is different from variable with name 'a'. (Difference in the upper-case and lower-case holds true)**

Printing/Providing output:

For printing statements in C++ programs, we use the cout statement.

For example: If you want to print "Hello World!" (without parenthesis) in your code, we will write it in following way:

```
cout << "Hello World!";
```

A full view of the basic C++ program is given below for the above example: Code:

```
#include <iostream>
using namespace std;
 int main()
 {
 cout<<"Hello World"
```

```
Output: Hello World!
```

Line separator:

For separating different lines in C++, we use endl or '\n'.

For example: Code:

```
#include <iostream>
using namespace std;
 int main()
 {
 cout<<"Hello World1"<<endl;
cout<<"Hello World2"<<'\n';
}
```

Output:

```
Hello World1
Hello World2
```