STATEMENTS IN C++

* Smallest executable unit of a programming language.
* Uses semi colon (;) as delimiter for the statement.
* Different types of statements in c++ includes.

1. Declaration statements
2. Assignment statements
3. Input statements.
4. Control statements.

1. DECLARATION STATEMENT

* variables should be declared before it is used in the program.
* variables are user defined word and its considered as an identifier.

Syntax

Data-type <variable 1>, <variable 2>, <variable 3> .....

* Each variables should be separated by comma (,).

VARIABLE INITIALISATION

* Assigning a value to variable during variable declaration is variable initialization.

Eg:- int a=5, b=20;

* value can be stored in the variable either during compilation or during the execution of the program.

const - The access modifier

* The keyword const is used to make a variable to a constant value. ie the value of such variables cannot be changed during the execution.

Eg:- const float p = 3.14;

* The above variable declaration the value of variable 'p' is initialized to 3.14 with the keyword const as a result the value of variable 'p' cannot be changed throughout the program.

2. ASSIGNMENT STATEMENT

* Assignment operator is used to assign values to a variable.

Eg :-

A = 15

r = sqrt(a) → parameter/argument

c = a+b

c = a

TYPE COMPATIBILITY

* The 2 different possibilities that occuros then an assignment statement is executed.

1. Size of the data type of the variable at LHS is higher than the variable or expression on the RHS. In this case data type of the value at the RHS is promoted (type promotion) to that of the variable at LHS.

```
#include<iostream.h>
Void main()
{
Int a=5, b=6;
Float c;
c=a;
cout<<c;
c=a+b;
cout<<c;
}
```

output

5.0 11.0

2. Size of the data type of the variable at LHS is smaller than the size of RHS value.

```
#include<iostream.h>
void main()
{
Float a=5.3, b=3.4;
Int c;
c=a;
cout<<c;
c=a+b
cout<<c;
}
```

output

58

3. INPUT STATEMENTS

* The keyword cin followed by getfrom or extraction operator (>>) specifies the input operation.

Eg :-

```
cin>>a>>b;
```

4. OUTPUT STATEMENTS

* The keyword cout followed by put to or insertion operator (<<) specifies the output operation.

Eg :-

```
cout<<a<<b;
```

CASCADING OF I/O OPERATORS

* The statement

```
cin>>a;
cin>>b;
cin>>c;
```

The above 3 statements can be combined to and rewritten into a single statement.

```
cin>>a>>b>>c;
```

and this called cascading of I/O operators.

★ STRUCTURE OF C++ PROGRAM

1. Preprocessor directives - preprocessor directive #include is used to link the header file in C++.

2. Header files - contains information about the keywords or predefined functions or predefined data types.

3. Main( ) function - execution starts from main( ) function and ends with in main function. Each statement is delimited by a semi colon (;)

1. PREPROCESSOR DIRECTIVE

* Preprocessor directive starts with #.
* Preprocessor directive #include is used to link the header files available in C++.
* Other preprocessor directives are #define, #undefine etc....

2. HEADER FILES

* Header files contain information like predefined data types, functions.
  Eg:- #include<iostream.h> contains the information about cin and cout.

3. Main() FUNCTION

* Execution of C++ program starts from the main() function and ends with in main().
* Each statement is delimited by semi colon (;).

★ GUIDELINES FOR CODING

* Use suitable names for identifiers.
* Use clear and simple expressions.
* Use comments when needed.
  - comment statements are used for internal documentation.
  - comment statements are not executed by the compiler.
  - comment statement are used to describe about the programs or the purpose of each statement used with in the program.
* comment the instructions in the program that are difficult to understand.
* comment the program instruction while writing the programs.
* write short and clear comments.

SINGLE LINE COMMENTS: // (double slash) is used as single line comment to comment in each line of programming instructions.

MULTILINE COMMENTS : if the description of the program exceeds to more than one line we can use multiline comments /* and */. multiline comments starts from /* and ends with */.

Instructions for using comments.

* comments in beginning of the program should describe the purpose of the programm.
* comment each variable declarations.

1. Write a c++ program area and perimeter of the circle of given radius. (area = Πr^2, perimeter = $2\Pi r$)

Ans)

Ans)
```
#include<iostream.h>
void main()
{
int r, area, peri; -> float pi=3.14;
cout<<"enter the radius";
cin>>r;
area=pi*r*r;
peri=2*pi*r;
cout<<"area="<<area;
cout<<" perimeter="<<peri;
}
```

2. Write a c++ program to find the simple interest (p*n*r/100).

Ans)

```
#include<iostream.h>
void main()
{
int p,n,r,amount;
cout<<"enter the principal amount";
cin>>p;
cout<<"enter number of years";
cin>>n;
cout<<"enter rate of interest";
cin>>r;
amount=p*n*r/100;
cout<<amount;
}
```

3. Write a C++ program to convert temperature from celsius to fahrenheit (fahrenheit = 1.8* celsius + 32)

Ans)

```
#include<iostream.h>
void main()
{
float c,f;
cout<<"enter the celsius";
cin>>c;
f=1.8*c+32;
cout<<"fahrenheit="<<f;
}
```