

SYNTAXES

1. variable declaration \rightarrow \langle datatype \rangle
 \langle variable name \rangle ;
2. Comma separating the individual variable \rightarrow \langle data type \rangle \langle var 1 \rangle , \langle var 2 \rangle ,
 \langle var 3 \rangle ... \langle var n \rangle ;
3. References \rightarrow \langle type \rangle \langle reference variable \rangle = \langle original variable \rangle ;
4. Setw formatting output \rightarrow setw (number of characters)
5. set precision \rightarrow set precision (number of digits) ;
6. Explicit type conversion? \langle type-name \rangle expression ;
7. If statement \rightarrow if (expression)
then - block ;
statement - x ;
8. If - else statement \rightarrow if (expression)
{
then - block ;
}
else
{
false - block ;
}
statement - x

7. nested if \rightarrow if (expression = 1)
 {
 if (expression = 2)
 {
 True part statement;
 }
 else
 {
 False part statement;
 }
 }

10. if else if ladder
 if (expression 1)
 {
 statement - 1
 }
 else
 if (expression 2)
 {
 statement - 2
 }
 else
 if (expression 3)
 {
 statement - 3
 }
 else
 {
 statement - 4
 }

11. Switch -> Switch (expression)

```

{
  case constant 1 ;
  statement (s) ;
  break ;
  case constant 2 ;
  statement (s) ;
  break ;
  .
  .
  .
  default ;
  statement (s) ;
}

```

12. for loop -> for initialization ; test - expression
update expression

```

{
  statement - 1 ;
  statement - 2 ;
  .
  .
  .
  statement - x ;
}

```

13. while - loop -> while (Test expression)

```

{
  Body of the loop ;
}
statement - x ;

```

14. Do while loop -> do

```

{
  Body of the loop ;
}
while (condition) ;

```


15. nested loop →
For (initialization ; test expression ;
update - expression (s) ;
{
 For initialization ; test expression ;
 update expression (s)
 {
 Statement (s) ;
 }
}

16. goto statement → goto label ;

label ;

17. strcpy → strcpy (target string, source string)

18. strlen → strlen (string)

19. strcmp → strcmp (string 1, string 2)

20. strcpyr → strcpyr (string)

21. strlenr → strlenr (string)

22. function definition → return data type
function name
(parameter list)
{

Body of the function

23. Constant argument → `<return type>`
`<function name>`
`<constant>`
`<datatype variable = value>`

24. Inline function: `<inline return type>`
`function name (data type`
`parameter 1, datatype`
`Parameter n)`

25. Return Statement → `return expression/variable`

26. one-dimensional array → `<data type>`
`<array name>`
`[array size]`

27. Initialization of 1-D array →
`<data type> <array name> [size] =`
`{ value - 1, value - 2, ..., value - n }`

28. 2-D array declaration → `data type array name [row-size]`
`[col-size]`

29. Declaring Structures → `struct Structure name`
`{`
`type number_name 1;`
`type number_name 2;`
`}`

`reference - name;`

30. class definition

```

class - class name
{
private:
    variable declaration,
    Function declaration,
protected:
    variable declaration,
    Function declaration,
public:
    variable declaration,
    function declaration,
};

```

31. outside the class -

```

return_type class_name::
function_name ( parameters list )
{
    Function definition
}

```

32. Calling member function

```

object_name . function_name
( actual parameters )

```

Kindl send me your district Questions & Keys to email Id - Padasalai.net@gmail.com