

# Classes & Objects.

## \* Defining Class :-

- A class is a group of objects which have common properties.
- It is a template from which objects are created.
- It is a logical entity.
- Once the class type has been defined, we can create "variables" of that type using declarations.

In Java, these variables are termed as instances of classes, which are the actual objects.

It includes :-  
= = = = =

- (i) Modifiers :- A class can be public or has default access.
- (ii) class keyword :- class keyword is used to create a class.
- (iii) class name :- The name of the class.
- (iv) fields
- (v) Methods
- (vi) Constructors
- (vii) Blocks

## Syntax

```
class classname  
{  
    [fields declaration ; ]  
    [methods declaration ; ]  
}
```

Note :- Everything inside the square brackets is optional.

\*

## Field Declaration

- o A Java field is a variable. This means that it represents a value, such as a numerical value or a text.
- o These variables are called instance variables.
- o fields are declared within classes.

## Example

```
class Rectangle  
{  
    int length, width;  
}
```

Instance variables

# \* Methods Declaration

- A Java Method is a function. It is a block of code that carries out an operation.
- Java Methods must be declared inside a classes, but immediately after the declaration of instance variables.

## Syntax

```

= = =
type methodname ( Parameter list )
    {
    method-body;
    }
  
```

## Example :-

```

= =
void show()
{
  System.out.println("JPwebdevelopers");
}
  
```

## Example of field and Method

```

= = = = =
class ABC
{
  int x;
  void method()
  {
  }
}
  
```



## \* Creating Objects :-

- An entity that has state and behaviours is known as an object.
- A Object is an instance of a class.
- Objects in java are created using the new operator.
- new operator :- It creates an object of the specified class and returns a reference of that object.

Syntax :- `classname object = new classname();`

## \* Accessing Class Members :-

All variables must be assigned values before they are used. Since we are outside the class, we cannot access the instance variables and the methods directly.

To access the class members, we must use the concerned object

and the dot operator.

Syntax: objectname.variablename = value;

objectname.methodname (parameter - list);

## # Application of Classes and Objects :-

```
class Student
```

```
{
```

```
int id = 101;
```

```
void show()
```

```
{
```

```
System.out.println("jwebdevelopers");
```

```
}
```

```
}
```

```
class Main
```

```
{
```

```
public static void main (String args [])
```

```
{
```

```
Student s1 = new Student ();
```

```
System.out.println (s1.id);
```

```
s1.show();
```

```
}
```

```
}
```