

Class & Object

Class & Object:



Class is a group of elements having common properties and behaviours.

- Class is virtual
- Object is real

Syntax:

- `Person obj=new Person();`

Type:

- User-defined eg.A,Person,Animal
- Pre-defined eg.System,string,Scanner

Program:

```
class Person { // my class name is Person
    int age = 20;
    int weight = 56;
    String color = "Light";

    void eat() {
        System.out.println("i am eating");
    }

    void sleep() {
        System.out.println("i am sleeping");
    }

    public static void main(String[] args) {
        Person object = new Person(); // declare class object
        System.out.println("Age is: " + object.age);
        System.out.println("Weight is: " + object.weight);
        System.out.println("color is: " + object.color);
        object.eat(); // called out method
        object.sleep();
    }
}
```

OUTPUT:-

```
Age is: 20
Weight is: 56
color is: Light
```

```
i am eating  
i am sleeping
```

constructor:

- constructor is a special type of method whose name is same as class name.
- The main purpose of constructor is initialize the object
- Every java class has a constructor
- A constructor is automatically called at the line of object creation
- A constructor never contain any return-type including void

Type:

- Private constructor
- Default constructor
- Copy constructor
- Parameterized constructor

1.Default Constructor

- A constructor which does not have any parameter is called default constructor

Syntax:

```
class Person {  
    Person(){  
        //code  
    }  
}
```

Program:

```
class Person {  
    int age;  
    String name;  
  
    Person() {  
        age = 20;  
        name = "Nikesh kumar";  
    }  
  
    void show() {  
        System.out.println("age is: " + age + " " + "and name is: " + name);  
    }  
}  
  
class A {  
    public static void main(String[] args) {
```

```

        Person object = new Person();
        object.show();
    }
}

```

OUTPUT:- age is: 20 and name is: Nikesh kumar

2.Parametrized Constructor

- constructor A through which we can pass one or more parameters is called parametrized constructor

Syntax:-

```

class Person {
    Person(int num1,int num2){
        //code
    }
}

```

Program:

```

class Person {
    int a, b;
    String fname, lname;

    Person(int num1, int num2) {    //parameter pass
        a = num1;
        b = num2;
    }

    Person(String firstName, String lastName) {
        fname = firstName;
        lname = lastName;

        System.out.println("full name is: " + firstName.concat(lastName));
    }

    void show() {
        System.out.println("sum of two number is: " + (a + b));
    }
}

class A {
    public static void main(String[] args) {
        Person object = new Person(10, 20); //initial value
        Person object2 = new Person("Sanjit", " kumar");
        object.show();
    }
}

```

OUTPUT:-
full name is: Sanjit kumar

```
sum of two number is: 30
```

3. Copy Constructor

- Whenever we pass object reference to the constructor then it is called copy constructor.

Syntax:

```
class Person {  
    Person(object ref){  
        //code  
    }  
}
```

Program:-

```
class Person {  
    int a;  
    String fname;  
  
    Person() {  
        a = 100;  
        fname = "avit";  
        System.out.println(fname + " " + a);  
    }  
  
    Person(Person refPerson) {    //pass our ref  
        a = refPerson.a;  
        fname = refPerson.fname;  
        System.out.println(fname + " " + a);  
    }  
}  
  
class A {  
    public static void main(String[] args) {  
        Person object = new Person();  
        Person object2 = new Person(object);  
    }  
}  
  
OUTPUT:  
avit 100  
avit 100
```

4.Private Constructor

- In java it is possible to write a constructor as a private but according to the rule we can not access member outside of class.

Syntax;

```
class Person {
```

```
private Person(){  
    //code  
}  
}
```

Program:

```
class Person {  
    int a;  
    String fname;  
  
    private Person() { // private constructor  
        a = 100;  
        fname = "avit";  
        System.out.println(fname + " " + a);  
    }  
  
    public static void main(String[] args) {  
        Person object = new Person();  
    }  
}  
OUTPUT:- avit 100
```