**Method return type**

**Primitive type :**

byte

short

int

long

float

double

char

boolean

int meth() {

return 5; // return statement

}

Example :

**class Animal1 {**

**int age;**

**int display(int n) {**

**age=n;**

**return age;**

**}**

**}**

**public class Demo1 {**

**public static void main(String[] args) {**

**Animal1 aa=new Animal1();**

**int value=aa.display(20);**

**System.*out*.println("the value is : "+value);**

**}**

**}**

**Constructor**

**Constructor is used to initialize the object.**

**Constructor name and class name should be same.**

**Constructor will call at the time of creating an object.**

**Constructor don’t have any return type not even void.**

**Constructor mostly used to initialize the instance variable.**

**There are two types of constructor**

* **Default constructor**
* **Parameterized constructor**

**Example :**

**class Car {**

**int model;**

**Car() { //default constructor**

**model=10;**

**}**

**Car(int a,int b) { // parameterized constructor**

**model=a+b;**

**}**

**void display() {**

**System.*out*.println("value of model is : "+model);**

**}**

**}**

**public class ConstDemo {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**Car cc=new Car();**

**Car cc1=new Car(5,20);**

**cc.display();**

**cc1.display();**

**}**

**}**

**Method Overloading :**

With in the class we can have more than one method with same name and with different

Parameter is called method overloading.

Based upon the argument that we passed to the parameter it decide which method to call.

It comes with compile time polymorpshim

**Example :**

**class Machine {**

**int tyre(int a) {**

**return a;**

**}**

**int tyre(int a,int b) {**

**return a+b;**

**}**

**int tyre(int a,int b,int c) {**

**return a+b+c;**

**}**

**}**

**public class MethodOverloading {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**Machine mm=new Machine();**

**System.*out*.println(mm.tyre(10));**

**System.*out*.println(mm.tyre(10,20));**

**System.*out*.println(mm.tyre(10,20,30));**

**}**

**}**

**Inheritance :**

**One object acquiring the properties from another object.**

**One class will inherit the properties from another class**

**The inherited class is a super class and inheriting class is a sub class**

**Single Inheritance**

**class A { // super class**

**-------**

**-------**

**}**

**// sub class**

**Class B extends A {**

**---------**

**---------**

**}**

**Class B inherits the members from class A using the extends keyword**

**Example :**

**class Father {**

**int i;**

**void display() {**

**System.*out*.println("printing super class variable : "+i);**

**}**

**}**

**class Child extends Father {**

**int j;**

**void display1() {**

**System.*out*.println("printing sub class variable : "+j);**

**}**

**}**

**public class InheritDemo {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**Child c=new Child();**

**c.i=10;**

**c.display();**

**c.j=20;**

**c.display1();**

**}**

**}**

**MultiLevel Inheritance :**

**It is a chain of inheritance**

**One class will inherit one class, that class will inherit another class.**

**class A {**

**//----**

**}**

**class B extends A {**

**//----**

**}class C extends B {**

**// ---**

**}**

**Example :**

**class A {**

**A() {**

**System.*out*.println("A class constructor..");**

**}**

**}**

**class B extends A {**

**B() {**

**System.*out*.println("B class constructor");**

**}**

**}**

**class C extends B {**

**C() {**

**System.*out*.println("C class constructor");**

**}**

**}**

**public class Multilevel {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**C a1=new C();**

**}**

**}**

**Java does not support multiple inheritance concept.**

**Super keyword :**

1. **To call the super class constructor.**
2. **To access the super class member.**

**Example 1 :**

**class A1 {**

**int b;**

**A1() {**

**System.*out*.println("this is A1 class constructor");**

**}**

**A1(int a) {**

**b=a;**

**System.*out*.println("this is super class parameterized constructor");**

**}**

**}**

**class B1 extends A1 {**

**B1() {**

**super(5);**

**System.*out*.println("this is B1 class constructor");**

**}**

**}**

**public class SuperDemo {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**B1 aa=new B1();**

**System.*out*.println("value of super class b : "+aa.b);**

**}**

**}**

**Example 2 :**

**class A2 {**

**int i=10;**

**void meth() {**

**System.*out*.println("cricket");**

**}**

**}**

**class B2 extends A2 {**

**int i=20;**

**void meth1() {**

**System.*out*.println("value of i is :"+i);**

**System.*out*.println("value of super class is : "+super.i);**

**super.meth();**

**}**

**}**

**public class Super1Demo {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**B2 b1=new B2();**

**b1.meth1();**

**}**

**}**