**public** **class** abstraction {

**private** **static** **int** *acno*=12345;

**private** **static** String *name*="lakshman";

**private** **static** **float** *balance*=10000;

**private** **float** profit=1000;

**private** **float** loan=100;

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

System.*out*.println("acno="+ *acno*);

System.*out*.println("name="+ *name*);

System.*out*.println("balance=" +*balance*);

}

}

**class** hash {

String name;

**int** age;

**void** talk()

{

System.*out*.println("hello iam"+ name);

System.*out*.println("my age is"+ age);

}

}

**public** **class** hashcode

{

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

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

hash h=**new** hash();

System.*out*.println(" h obj hashcode is:"+ h.hashCode());

}

}

**import** java.text.NumberFormat;

**public** **class** factorymethodandfinalkeyword {

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

**final** **double** PI=(**double**)22/7;

**double** r=15.5;

**double** area=PI\*r\*r;

System.*out*.println("area is:="+area);

NumberFormat obj=NumberFormat.*getNumberInstance*();

obj.setMaximumFractionDigits(2);

obj.setMinimumIntegerDigits(7);

String str=obj.format(area);

System.*out*.println("formated area is="+str);

}

}

**class** Sup{

String name="pavan";

**int** id=1;

**void** get(){

System.*out*.println("super class "+name);

System.*out*.println("super class "+id);

}

}

**class** Sub **extends** Sup{

String addr="hyd";

**void** get(){

Sup s=**new** Sup();

s.get();

System.*out*.println("sub class "+addr);

}

}

**public** **class** InheritanceDemo {

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

Sub s2=**new** Sub();

s2.get();

}

}

//overloading means WRITING TWO OR MORE METHODS IN SINGLE CLASS

**class** Onea{

**void** calc(**int** x,**int** y){

System.*out*.println("square value of "+(x\*x));

System.*out*.println("square value of "+(y\*y));

}

**void** calc(**int** x,**int** y,**int** z){

System.*out*.println("square value of "+(x\*x));

System.*out*.println("square value of "+(y\*y));

System.*out*.println("square value of "+(z\*z));

}

}

**public** **class** Overloading {

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

Onea o1=**new** Onea();

o1.calc(1, 2);

o1.calc(3,4,5);

}

}

**class** One0{

**void** calc(**int** x){

System.*out*.println("square value of "+(x\*x));

}

}

**class** Two0 **extends** One0{

**void** calc(**int** x){

System.*out*.println("square value of "+(x\*x\*x));

}

}

**public** **class** Overriding2 {

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

One0 o1=**new** One0();

Two0 o2=**new** Two0();

One0 ref;

//ref=o1;

//ref.calc(2);

o1.calc(50);

ref=o2;

ref.calc(2);

}

}

// Programs to print various data

**class** ints

{

**int** a=10; **int** b=20; **int** c=30;

**void** display()

{

System.*out*.println("\nInteger Value of a : "+a);

System.*out*.println("Integer Value of b : "+b);

System.*out*.println("Integer Value of c : "+c);

}

}//ints

**class** doubles

{

**double** a=10.11; **double** b=20.22; **double** c=30.33;

**void** display()

{

System.*out*.println("\n\nDouble Value of a : "+a);

System.*out*.println("Double Value of b : "+b);

System.*out*.println("Double Value of c : "+c);

}

}//doubles

**class** chars

{

**char** a='@'; **char** b='Z'; **char** c='\*';

**void** display()

{

System.*out*.println("\n\nchar Value of a : "+a);

System.*out*.println("char Value of b : "+b);

System.*out*.println("char Value of c : "+c);

}

}//chars

**class** saticpolymorphism

{

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

{

ints i=**new** ints();

i.display();

doubles d=**new** doubles();

d.display();

chars c=**new** chars();

c.display();

}

}

**class** One1{

**int** i=10;

**void** show(){

System.*out*.println("Super class method of i "+i);

}

}

**class** Two2 **extends** One1{

**int** i=20;

**void** show(){

System.*out*.println("Sub class method of i "+i);

**super**.show();

System.*out*.println("Super "+i);

}

}

**public** **class** SuperKWord {

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

Two2 o=**new** Two2();

o.show();

}

}

**class** Bc

{

Bc()

{

System.*out*.println("Bc-Dc");

}

Bc(**int** x)

{

**this**();

System.*out*.println("Bc-Pc");

}

}

**class** Ibc **extends** Bc

{

Ibc()

{

**super**(1000);

System.*out*.println("Ibc-Dc");

}

Ibc(**int** x)

{

**this**();

System.*out*.println("Ibc-Pc");

}

}

**class** Dc **extends** Ibc

{

Dc()

{

**super**(100);

System.*out*.println("Dc-Dc");

}

Dc(**int** x)

{

**this**();

System.*out*.println("Dc-Pc");

}

}

**class** SuperThisDemo

{

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

{

Dc ob=**new** Dc(1);

}

}

**public** **class** thiskeyword {

**private** **int** x;

thiskeyword(){

**this**(55);

**this**.access();

}

thiskeyword(**int** x){

**this**.x=x;

}

**public** **void** access()

{

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

}

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

thiskeyword tw=**new** thiskeyword();

}

}

**class** instance1 {

String name;

**int** age;

**void** talk()

{

System.*out*.println("my name is:"+ name);

System.*out*.println("my age is:"+ age);

}

}

**class** instancevariable

{

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

instance ist=**new** instance();

ist.name="lakshman";

ist.age=25;

ist.talk();

}

}

**class** staticvrb {

**static** **int** *i*=50;

**static** **void** access()

{

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

}

}

**class** staticvariable

{

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

staticvrb.*access*();

}

}

//static methods are not act upon the instance variable of a class.

//this methods called by CLASS NAME

//static variables are stored in method area

**class** Test{

//instance variable

**static** **int** *x*=9;

//static variables accessed by static methods only

/\*Test(int x){

this.x=x;

}\*/

**static** **void** access(){

System.*out*.println("x="+*x*);

}

}

**public** **class** StaticMethods {

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

//Test 0=new Test();

Test.*access*();

}

}

**class** constructor2 {

**static** **double** sum(**double** num1,**double** num2)

{

**double** res=num1+num2;

**return** res;

}

}

**class** staticmethod

{

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

constructor2 cont=**new** constructor2();

**double** d=constructor2.*sum*(10,20.25);

System.*out*.println("sum is=" +d);

}

}

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

class instances {

private String name;

private int age;

public void accept()throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("enter name:");

name=br.readLine();

System.out.println("enter age:");

age=Integer.parseInt(br.readLine());

}

public void check()throws IOException

{

if(age<=30)

System.out.println(name + ":is young");

else if(age<=50)

System.out.println(name +":is middle aged");

else System.out.println(name +":is old");

}

}

class instancemethod1

{

public static void main(String[] args)throws IOException {

instances inst=new instances();

inst.accept();

inst.check();

}

}

**class** instance {

String name="lakshman";

**int** age=25;

**void** talk()

{

System.*out*.println("my name is:"+ name);

System.*out*.println("my age is:"+ age);

}

}

**class** instancemethod

{

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

instance ist=**new** instance();

ist.talk();

}

}

//converting one data type to another data type is called type casting

//widenning: converting lower to up

//narowinging: converting up to lower

**class** Onet{

**int** i=10;

**void** show(){

System.*out*.println("Super class method of i "+i);

}

}

**class** Twot **extends** Onet{

**int** i=20;

**void** show(){

System.*out*.println("Sub class method of i "+i);

}

}

**public** **class** TypeCasting {

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

Onet o;//o is a super class reference

o=(Onet)**new** Twot();//o is referring to sub class object

//widening

o.show();//sub clas of method of i

/\*

// class cast Exception

Twot t;//t is sub referenc

t=(Twot)new Onet();

//the above is narrowing

t.show();\*/

}

}