**Method Overloading**

1) Calculate Area of rectangle, circle, triangle

import java.util.\*;

class Poly{

double l,b,A,r,h;

void cal\_area(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter l & b");

l=sc.nextDouble();

b=sc.nextDouble();

A=l\*b;

System.out.println("Area of rectangle="+A);

}

void cal\_area(double r){

this.r=r;

A=3.14\*r\*r;

System.out.println("Area of circle="+A);

}

double cal\_area(double b, double h){

this.b=b;

this.h=h;

A=0.5\*b\*h;

return A;

}

}

public class Main

{

public static void main(String[] args) {

double r,b,h;

Poly p=new Poly();

p.cal\_area();

Scanner sc=new Scanner(System.in);

System.out.println("Enter radius of circle");

r=sc.nextDouble();

p.cal\_area(r);

System.out.println("Enter b & h");

b=sc.nextDouble();

h=sc.nextDouble();

System.out.println("Area of triangle="+p.cal\_area(b,h));

}

}

O/P:

Enter l & b

12

2

Area of rectangle=24.0

Enter radius of circle

2.2

Area of circle=15.197600000000003

Enter b & h

12

34

Area of triangle=204.0

2) max 2 number & max 3 number using method overloading

int max(int a,int b) void max()

import java.util.\*;

class MaxNumber{

int a,b;

int max(int a,int b){

this.a=a;

this.b=b;

if(a>b){

return a;

}

else{

return b;

}

}

void max(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter value of a & b");

a=sc.nextInt();

b=sc.nextInt();

if(a>b){

System.out.println(a+" is max");

}

else if(b>a){

System.out.println(b+" is max");

}

else{

System.out.println("Both are equal & max");

}

}

}

public class Main

{

public static void main(String[] args) {

int a,b;

Scanner sc=new Scanner(System.in);

System.out.println("Enter value of a & b");

a=sc.nextInt();

b=sc.nextInt();

MaxNumber mn=new MaxNumber();

System.out.println(mn.max(a,b)+" is Max");

mn.max();

}

}

O/P:

Enter value of a & b

12

13

13 is Max

Enter value of a & b

21

10

21 is max

3) Add 2 number & Add3 number using method overloading

import java.util.\*;

class Addition{

int a,b;

void add(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter value of a & b");

a=sc.nextInt();

b=sc.nextInt();

System.out.println("Addition="+(a+b));

}

void add(int a,int b){

this.a=a;

this.b=b;

System.out.println("Addition="+(a+b));

}

}

public class Main

{

public static void main(String[] args) {

int a,b;

Addition addition=new Addition();

addition.add();

Scanner sc=new Scanner(System.in);

System.out.println("Enter value of a & b");

a=sc.nextInt();

b=sc.nextInt();

addition.add(a,b);

}

}

O/P:

Enter value of a & b

12

10

Addition=22

Enter value of a & b

34

32 87

Addition=121

Method Overriding

1) Animal

import java.util.\*;

class Animal{

void eat(){

System.out.println("Eating....");

}

}

class Dog extends Animal{

void eat(){

System.out.println("Dog Eating Bread");

}

}

class Cow extends Animal{

void eat(){

System.out.println("Cow Eating Grass");

}

}

public class Main

{

public static void main(String[] args) {

Animal a=new Animal();

a.eat();

Animal d=new Dog();

d.eat();

Cow c=new Cow();

c.eat();

}

}

O/P:

Eating....

Dog Eating Bread

Cow Eating Grass

2) Shape

class Shape{

public void draw(){

System.out.println("drawing...");

}

}

class Rectangle extends Shape{

public void draw(){

//Logic(l,b);

System.out.println("drawing rectangle...");

}

}

class Circle extends Shape{

public void draw(){

//logic(r);

System.out.println("drawing circle...");

}

}

public class Main{

public static void main(String[] args){

Shape s;

s=new Shape();

s.draw();

s=new Rectangle();

s.draw();

s=new Circle();

s.draw();

}

}

O/P:

drawing...

drawing rectangle...

drawing circle...

3) Area & volume

import java.util.Scanner;

class Area

{

double r,A;

Area(double r)

{

this.r=r;

}

void cal\_area()

{

A=3.14\*r\*r;

System.out.println("Radius="+r+"\nArea="+A);

}

}

class Volume extends Area

{

double h,v;

Volume(double r,double h)

{

super(r);

this.h=h;

}

void cal\_area()

{

super.cal\_area();

v=A\*h;

System.out.println("H="+h+"\nVolume="+v);

}

}

public class Main

{

public static void main(String[] args)

{

double r,h;

Scanner sc=new Scanner(System.in);

System.out.println("Enter r & h");

r=sc.nextDouble();

h=sc.nextDouble();

Volume v= new Volume(r, h);

v.cal\_area();

}

}

O/P:

Enter r & h

12

3.5

Radius=12.0

Arae=452.15999999999997

H=3.5

Volume=1582.56

4) Date, Employee, Manager, SalesManager

class Date {

int dd; int mm; int yy;

public Date(){ dd=mm=yy=0; }

public Date(int d,int m,int y){

dd=d; mm=m; yy=y;

}

public String toString() {

return dd+"/"+mm+"/"+yy;

}

}

class Employee {

int empID; String ename; Date bdate;

int wdays;// working days in month

double rate; //rate per day

public Employee() {}

public Employee(int eid,String n, Date d, int wd,double r){

empID=eid; ename=n; bdate=d; wdays=wd; rate=r;

}

}

class Manager extends Employee {

double salary;

Manager() {

super(); salary=0;

}

Manager(int eid, String s,Date d,int wd, double rate) {

super(eid,s,d,wd,rate);

}

public double computesal(){

return (wdays\*rate);

}

public String toString() {

return empID+"\n"+ename+"\n"+bdate+"\n"+wdays+"\n"+rate+"\n"+this.computesal();

}

}

class SalesManager extends Manager{

double sales; double comm;

SalesManager(){

super();

sales=0;

comm=0;

}

SalesManager(int eid,String n, Date d,int wd,double r,double s,double c)

{

super(eid,n,d,wd,r);

sales=s; comm=c;

}

public double computesal(){

if (sales > 1000)

return(super.computesal()+sales\*comm);

else

return(super.computesal());

}

public String toString(){

return empID+"\n"+ename+"\n"+bdate+"\n"+wdays+"\n"+rate+"\n"+this.computesal();

}

}

public class Main{

public static void main(String[] args)

{

Date d1=new Date(14,7,1979);

Employee e1=new Employee(10,"A",d1,23,100.50);

Manager m1=new Manager(10,"B",d1,23,200.50);

System.out.println(m1);

Date d2=new Date(12,4,2000);

SalesManager sm1=new SalesManager(20,"C",d2,27,150,1500,10.5);

System.out.println(sm1);

}

}

O/P:

10

B

14/7/1979

23

200.5

4611.5

20

C

12/4/2000

27

150.0

19800.0