**Experiment 3a**

**Aim:**  WAP using the concept of method overloading to calculate and print the volume of different objects like Cube, Cuboid, Cylinder, Cone. Create multiple overloads for a method name which are differentiated by the number of their parameters. Input the details from the user and print the volume of the object

**Theory:** method overloading means using the same method name, but changing the data type of the parameters or the number of parameters to differentiate between the methods. Constructor overloading means same as method overloading as constructors are also essentially methods.

**Algorithm:**

1. public class Volume{ public static void main(String[] args){
2. sout(1.cube,2.cuboid,3.cylinder,4.cone)
3. while(true){ask user for their choice and store in int n
4. switch(n){case 1: ask for side of the cube and store in int a
5. volume v1 = new volume(a), vol(a);
6. case 2:ask for length breadth and height of the cuboid,volume v2 = new volume(l, b, h), vol(l,b,h);
7. case 3: ask for radius and height and store in r, s. volume v3 = new volume(r, s), vol(r,s)
8. case 4: ask for radius and height and store in x and i.volume v = new volume(x, i, 1), vol(x,i,1);
9. inside class volume{

**Code:**

import java.util.Scanner;

public class Volume {

public static void main(String[] args) {

System.out.println("1. Cube");

System.out.println("2. Cuboid");

System.out.println("3. Cylinder");

System.out.println("4. Cone");

while(true){

Scanner sc = new Scanner(System.in);

System.out.print("enter your choice: ");

int n = sc.nextInt();

switch(n) {

case 1:

System.out.print("enter side= ");

int a = sc.nextInt();

volume v1 = new volume(a);

vol(a);

break;

case 2:

System.out.print("enter length, breadth and height= ");

int l = sc.nextInt();

int b = sc.nextInt();

int h = sc.nextInt();

volume v2 = new volume(l, b, h);

vol(l, b, h);

break;

case 3:

System.out.print("enter radius and height= ");

double r = sc.nextDouble();

int s = sc.nextInt();

volume v3 = new volume(r, s);

vol(r, s);

break;

case 4:

System.out.print("enter radius and height= ");

double x = sc.nextDouble();

int i = sc.nextInt();

volume v = new volume(x, i, 1);

vol(x, i, 1);

break;

}

System.out.print("enter 0 to exit or any key to continue: ");

int g=sc.nextInt();

if(g==0){

break;

}

}

}

public static void vol(int a){

int vo=a\*a\*a;

System.out.println("the volume of the cube is "+vo);

}

public static void vol(int l, int b , int h){

int vo = l\*b\*h;

System.out.println("the volume of the cuboid is "+vo);

}

public static void vol(double r, int h){

double vo = 3.14\*r\*r\*h;

System.out.println("the volume of cylinder is "+vo);

}

public static void vol(double r, int h, int c){

double vo = 0.33\*3.14\*r\*r\*h;

System.out.println("the volume of cone is "+vo);

}

}

class volume{

int l,b,h;

double r;

volume(int a){

this.l=a;

}

volume(int l , int b, int h){

this.l=l;

this.b=b;

this.h=h;

}

volume(double r, int s){

this.r=r;

this.h=s;

}

volume(double r, int s, int c){

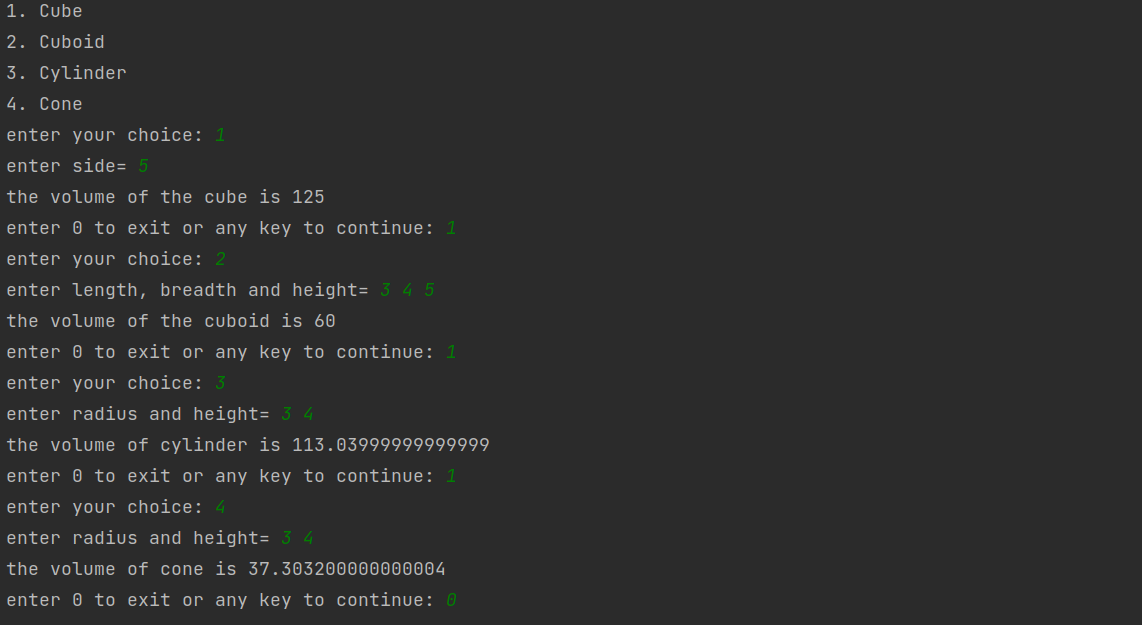
this.r=r;

this.h=s;

}

}

**Output:**

****

**Conclusion:**

after writing this program, I learnt how to use method and constructor overloading and understood its purpose.