**Class Objects as Parameters in Methods**

Similar to any other primitive parameter, an object of a class may also be a parameter of a method. The type of an object of a class is the name of its class. The operations that the class object may be subjected to are the ones derived from the class Object and those defined in its own class and the super class if it is the object of a derived class.

class Farm

{

double length;// declaration of variables

double width;

Farm (double l, double w)// method definition

{

length = l; width = w;

}

/\*Following is the method with parameter farm an object of Farm.\*/

void equals(Farm farm)

{// method definition

if (farm.length \* farm.width == length\*width)

System.out.println("True");

else

System.out.println("False");

}

}

class FarmExec4// class with main method

{

public static void main (String Str[])

{

Farm farm1 = new Farm (25, 10); //creating objects

Farm farm2 = new Farm(30, 20);

Farm farm3 = new Farm (15,40);

farm3. equals(farm1);// accessing methods

farm3. equals (farm2);

}

}

**Recursive Methods**

It is a technique of defining a method so that it calls itself in one of its statements.

Graphical user interface, text, application

Description automatically generated

**Nesting of Methods**

It implies a method calling another method in the same class. Since a method calls another method in the same class, dot(.) operator is not needed when it is called (Program 6.10). The following possibilities are allowed:

1. A method can call more than one method in the same class.

2. Successive calls can be made. This implies the first method calls the second method; the second method calls the third method, and so on.

import java.util.Scanner;

public class Rectangle

{

int perimeter(int l, int b)

{

int pm = 2\*(l + b);

return pm;

}

int area(int l, int b)

{

int pm = perimeter(l, b);

System.out.println("Perimeter is :" + pm);

int ar = l\*b;

return ar;

}

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

System.out.println("enter the length of rectangle:");

int l = s.nextInt();

System.out.println("enter the breadth of rectangle:");

int b = s.nextInt();

Rectangle obj = new Rectangle();

int ar = obj.area(l,b);

System.out.println("Area is :" +ar);

}

}