using System;

using System.Collections;

namespace ConsoleApp1

{

public class Program

{

//call by referenec using "ref" keyword:-

public void Show3(ref int val)

{

val += val; // Manipulating value

Console.WriteLine("Value inside the show function " + val);

// No return statement

}

public void Show(int val)

{

val += val; // Manipulating value

Console.WriteLine("Value inside the show function " + val);

// No return statement

}

public void Show2(out int val) // Out parameter

{

int square = 5;

val = square;

val \*= val; // Manipulating value

}

public int id;

public String name;

public float salary;

public Program(int i, String n, float s)

{

id = i;

name = n;

salary = s;

}

public Program()

{

}

~Program()

{

Console.WriteLine("Destructor Invoked");

}

public void display()

{

Console.WriteLine(id + " " + name + " " + salary);

}

//Abstaract class

public abstract class Shape

{

public abstract void draw();

}

public class Rectangle : Shape

{

public override void draw()

{

Console.WriteLine("drawing rectangle...");

}

}

public class Circle : Shape

{

public override void draw()

{

Console.WriteLine("drawing circle...");

}

}

//Interface

public interface Drawable

{

void draw();

}

public class Rectangle1 : Drawable

{

public void draw()

{

Console.WriteLine("drawing rectangle...");

}

}

public class Circle1 : Drawable

{

public void draw()

{

Console.WriteLine("drawing circle...");

}

}

public static void Main(string[] args)

{

// Create a set of strings

ArrayList list = new ArrayList(); // We are

list.Add(12);

list.Add(17);

list.Add(11);

list.Add(10);

list.Add(16);

list.Sort(); //we are sorting our array with the help of sort() method.

// Iterate SortedSet elements using foreach loop

Console.WriteLine("Array List ");

foreach (var name in list)

{

Console.WriteLine(name);

}

Console.WriteLine("Call By Value");

Console.WriteLine();

int val = 50;

Program program = new Program(); // Creating Object

Console.WriteLine("Value before calling the function " + val);

program.Show(val); // Calling Function by passing value

Console.WriteLine("Value after calling the function " + val);

Console.WriteLine("Call By Reference");

Console.WriteLine();

Program program2 = new Program();

Console.WriteLine("Value before calling the function " + val);

program2.Show3(ref val); // Calling Function by passing value

Console.WriteLine("Value after calling the function " + val);

Console.WriteLine("Call By Out Parameter");

Console.WriteLine();

Program program3 = new Program();

Console.WriteLine("Value before calling the function " + val);

program3.Show2(out val); // Calling Function by passing value

Console.WriteLine("Value after calling the function " + val);

Program e1 = new Program(101, "Sonoo", 890000f);

Program e2 = new Program(102, "Mahesh", 490000f);

Program e3 = new Program(); // Destructor

Console.WriteLine("Constructor");

Console.WriteLine();

e1.display();

e2.display();

Console.WriteLine("Destructor");

Console.WriteLine();

e3.display();

Console.WriteLine("Absrtact class");

Console.WriteLine();

Shape s;

s = new Rectangle();

s.draw();

s = new Circle();

s.draw();

Console.WriteLine("Interface");

Console.WriteLine();

Drawable d;

d = new Rectangle1();

d.draw();

d = new Circle1();

d.draw();

}

}

}

