Call by Value, reference and out parameter

using System;

namespace ConsoleApp7

{

class Program

{

public void show(out int val)

{

int val1 = 50;

val = val1;//20

val \*= val; //val = val \* val

Console.WriteLine("value inside the show method:" + val);//400

}

static void Main(string[] args)

{

int val = 20;

Program p1 = new Program();

Console.WriteLine("before calling the method" + val);//20

// p1.show(val);//call by value

// p1.show(ref val);//call by reference

p1.show(out val);

Console.WriteLine("after calling the method" + val);//400

}

}

}

Method Overloading

namespace ConsoleApp6

{

public class sample

{

public static float pi = 3.14f;

public static int cube(int n)//static method

{

return n \* n \* n;

}

public int add(int a,int b)//non static method - need object

{

return a + b;

}

public float add(float a,float b)

{

return a + b;

}

public string add(string a, string b)

{

return a + b;

}

public int add(int a,int b,int c)

{

return a + b + c;

}

}

class Program

{

public static void Main(string[] args)

{

sample s1 = new sample();

Console.WriteLine(s1.add(5,10,20));

Console.WriteLine(s1.add("good","evening"));

}

}

}

using System;

namespace ConsoleApp3

{

public class Employee

{

public int eid;

public string ename;

public float salary;

public Employee()

{

Console.WriteLine("Default constructor");

}

public Employee(int i,string n,float s)

{

eid = i;

ename = n;

salary = s;

}

public void display()

{

Console.WriteLine(eid + " " + ename + " " + salary);

}

}

class Program

{

public static void Main(string[] args)

{

Employee e1 = new Employee();

Employee e2 = new Employee(1, "Anu", 50000f);//initialization of member variables has been done during object instantiation

e2.display();

}

}

}