Properties

Inheritance

Dll

Class student

{

Int rn;

String name;

Public void Get() {}

Public void Display(){}

}

Properties : Wrapper around the attributes

Are used to access the private variables of a class outside the class

Do some validation on the values

Class student

{

Int rn;

Public int Rn { get ; set;}

String name;

Public string Name { get ; set;}

Public void Get() {}

Public void Display(){}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Student

{

int rn;

public int Rn

{

get

{

return rn;

}

set

{

rn =value;

}

}

string name;

public string Name

{

get

{

return name;

}

set

{

name = value;

}

}

int marks;

public int Marks

{

get

{

return marks;

}

set

{

marks = value;

}

}

string batch;

public string Batch

{

get

{

return batch;

}

set

{

batch = value;

}

}

}

}

Accessing Properties in Program. Cs file

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Program

{

static void Main(string[] args)

{

Student student = new Student();

student.Rn = 10;

Console.WriteLine("Roll No is " + student.Rn);

student.Name = "Ajay";

Console.WriteLine("Name is " + student.Name);

}

}

}

Validation

int rn;

public int Rn

{

get

{

return rn;

}

set

{

if (rn < 0)

rn = 0;

else

rn =value;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Student

{

int rn;

public int Rn

{

get

{

return rn;

}

set

{

rn = value;

}

}

string name;

public string Name

{

get

{

return name;

}

set

{

name = value;

}

}

int marks;

public int Marks

{

get

{

return marks;

}

set

{

marks = value;

}

}

string batch;

public string Batch

{

get

{

return batch;

}

set

{

batch = value;

}

}

public Student() { }

public Student(int rn , string name , string batch , int marks)

{

this.rn = rn;

this.name = name;

this.batch = batch;

this.marks = marks;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Program

{

static void Main(string[] args)

{

Student s = new Student();

Student s1 = new Student(1, "Ajay", "B1", 90);

Student s3 = new Student(2, "Bobby", "B2", 90);

Student[] st = new Student[10];

int rn, marks;

string name, batch;

for(int i=0;i<10;i++)

{

Console.WriteLine("EnterRollNo");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

name=Console.ReadLine();

Console.WriteLine("Enter Batch");

batch=Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Convert.ToByte(Console.ReadLine());

st[i] = new Student(rn, name, batch, marks);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Program

{

static void Main(string[] args)

{

////Student s = new Student();

////Student s1 = new Student(1, "Ajay", "B1", 90);

////Student s3 = new Student(2, "Bobby", "B2", 90);

Student[] st = new Student[10]

{

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90)

};

//int rn, marks;

//string name, batch;

//for(int i=0;i<10;i++)

//{

// Console.WriteLine("EnterRollNo");

// rn = Convert.ToByte(Console.ReadLine());

// Console.WriteLine("Enter Name");

// name=Console.ReadLine();

// Console.WriteLine("Enter Batch");

// batch=Console.ReadLine();

// Console.WriteLine("Enter Marks");

// marks = Convert.ToByte(Console.ReadLine());

// st[i] = new Student(rn, name, batch, marks);

//}

}

}

}

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Program

{

static void Main(string[] args)

{

//ArrayList list = new ArrayList();

List<Student> list = new List<Student>();

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

list.Add(new Student(1, "Ajay", "B1", 90));

////Student s = new Student();

////Student s1 = new Student(1, "Ajay", "B1", 90);

////Student s3 = new Student(2, "Bobby", "B2", 90);

// Student[] st = new Student[10]

// {

// new Student(1, "Ajay", "B1", 90),

// new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90)

// };

//int rn, marks;

//string name, batch;

//for(int i=0;i<10;i++)

//{

// Console.WriteLine("EnterRollNo");

// rn = Convert.ToByte(Console.ReadLine());

// Console.WriteLine("Enter Name");

// name=Console.ReadLine();

// Console.WriteLine("Enter Batch");

// batch=Console.ReadLine();

// Console.WriteLine("Enter Marks");

// marks = Convert.ToByte(Console.ReadLine());

// st[i] = new Student(rn, name, batch, marks);

//}

}

}

}

Collection Initializer

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Program

{

static void Main(string[] args)

{

// USE COLLECTION INITIALIZER WHEN TO STORE

//MULTIPLE OBJECTS OF SAME TYPE

List<Student> list = new List<Student>()

{

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90),

new Student(1, "Ajay", "B1", 90)

};

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

//list.Add(new Student(1, "Ajay", "B1", 90));

////Student s = new Student();

////Student s1 = new Student(1, "Ajay", "B1", 90);

////Student s3 = new Student(2, "Bobby", "B2", 90);

// Student[] st = new Student[10]

// {

// new Student(1, "Ajay", "B1", 90),

// new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90),

//new Student(1, "Ajay", "B1", 90)

// };

//int rn, marks;

//string name, batch;

//for(int i=0;i<10;i++)

//{

// Console.WriteLine("EnterRollNo");

// rn = Convert.ToByte(Console.ReadLine());

// Console.WriteLine("Enter Name");

// name=Console.ReadLine();

// Console.WriteLine("Enter Batch");

// batch=Console.ReadLine();

// Console.WriteLine("Enter Marks");

// marks = Convert.ToByte(Console.ReadLine());

// st[i] = new Student(rn, name, batch, marks);

//}

}

}

}

Collection Initializer using Properties

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp33

{

class Program

{

static void Main(string[] args)

{

// USE COLLECTION INITIALIZER WHEN TO STORE

//MULTIPLE OBJECTS OF SAME TYPE

List<Student> list = new List<Student>()

{new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90},

new Student() { Rn=1, Name="Ajay", Batch="B001", Marks=90}

};

}

}

}

Assembly : Unit of Deployment

It could be exe , dll

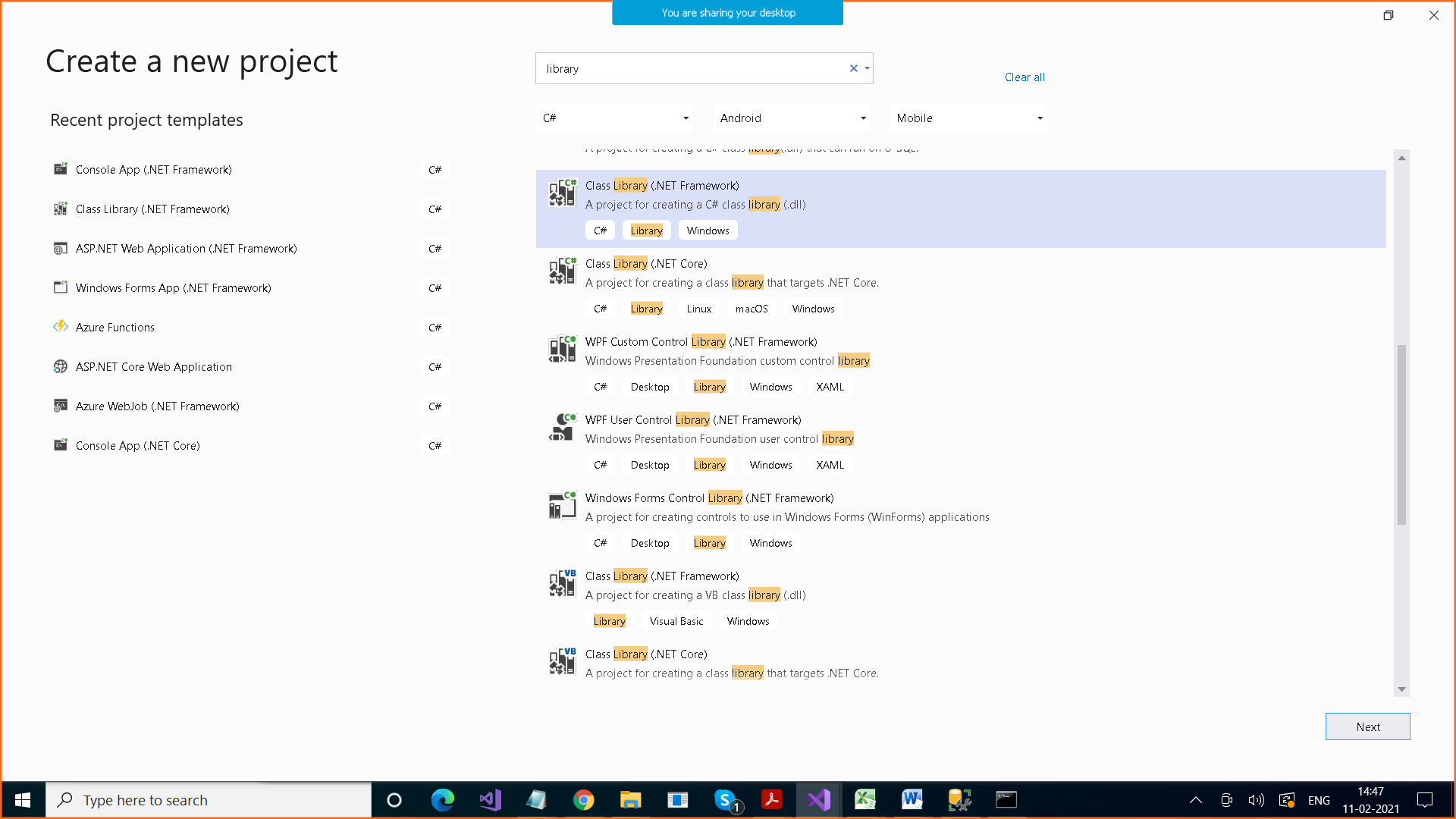
Exe >which can be executed (Console Application , Windows Application)

Dll > dynamic Link Library ( which can be executed ) (Class Library)

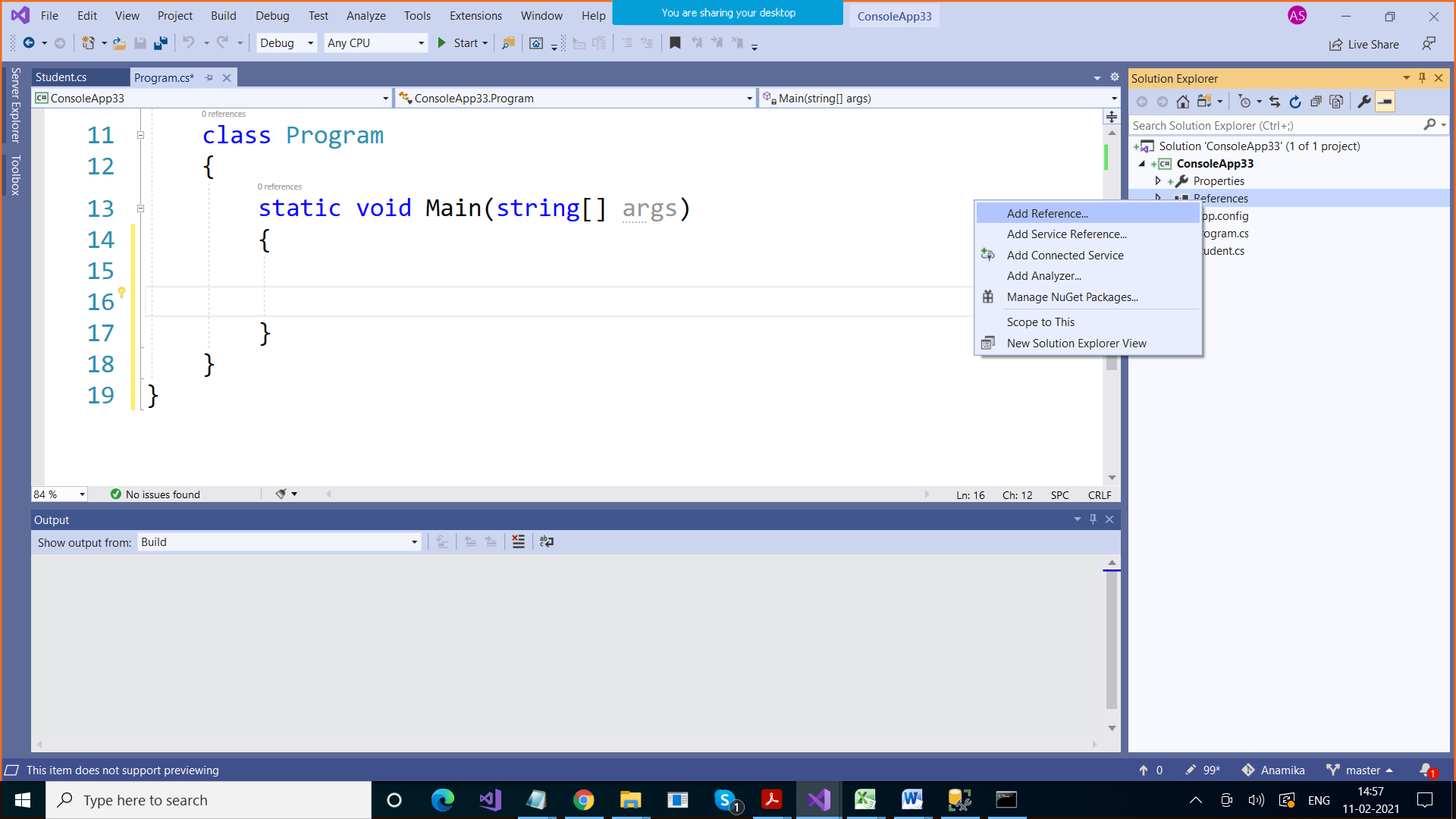
It’s a project which contains functions /methods only , which we can use in other projects

**REUSBALITY**

**Why do we make functions : Reusability**



**To refer to a dll, Add Reference**



E:\CTS\Myfunctions\Myfunctions\bin\Debug

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Myfunctions

{

public class StringFunctions

{

public string CombineStrings(string string1 , string string2)

{

return string1 + " " + string2;

}

public int StringLength(string string1)

{

return string1.Length;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Myfunctions

{

public class NumericFunctions

{

public int add(int x, int y)

{

return x + y;

}

public int subtract(int x, int y)

{

return x - y;

}

int product(int x, int y)

{

return x \* y;

}

int divide(int x, int y)

{

return x / y;

}

}

}

Build It > Don’t run it

When you build this project , we get dll

After that, wherever you want to use these methods , Add Reference , add dll

After that to call the methods , you need to create object of those classes