Day 10 C# ASP.NET

\*Class and Objects – inheritance, Abstract and Sealed

C# mathematical function

All mathematical function is store inside the library.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace math\_function

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(Math.Min(5,10));

Console.WriteLine(Math.Max(10, 15));

Console.WriteLine(Math.Sqrt(64));

Console.WriteLine(Math.Abs(-4.7));

Console.WriteLine(Math.Round(9.46));

Console.ReadLine();

}

}

}

Task 1:

Write a program to input two number form the user and perform following operation in user choice.

If choice = 1 – perform min function

Choice = 2 – max function

Choice = 3 – sqrt

Choice 4 – abs()

Choice 5 – round()

OOPS

Object oriented program.

Class and Object

Class is collection of member and methods.

Member: All variable inside class called members.

Method: All type of function inside the class called methods.

Types of class

1. User Class – create before the main class
2. Main class

How to create class?

Syntax –

Class name

{

Member

Method

}

In class and object cannot count main class.

By default class is called private.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace @class

{

class student

{

public int roll;

}

class Program

{

static void Main(string[] args)

{

student s1 = new student();

s1.roll = 10;

Console.WriteLine(s1.roll);

Console.ReadKey();

}

}

}

Method:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace @class

{

class student

{

public int roll;

public void diaplay()

{

Console.WriteLine("This us class and object");

}

}

class Program

{

static void Main(string[] args)

{

student s1 = new student();

s1.roll = 10;

Console.WriteLine(s1.roll);

s1.diaplay();

Console.ReadKey();

}

}

}

Class and ob**ject button up approach**

**Parameter using class and object**

**Assignment:**

**Create a class student**

Create a method input()

Except following parameter roll, name, city, branch, college, total, per,

Create method display ()

Print all info with specific condition

Per > 60 && branch = CSE && total >1 50

Otherwise print re admission

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace @class

{

class student

{

int groll, gper, gtot;

string gname, gcity, gbranch;

public void input(int nroll, string nname , string ncity, string nbranch, int ntotal, int npercent)

{

groll = nroll;

gname = nname;

gcity = ncity;

gbranch = nbranch;

gtot = ntotal;

gper = npercent;

display();

}

public void display()

{

if(gper > 60 && gbranch == "CSE" && gtot > 150)

{

Console.WriteLine("\nRoll " + groll +

"\nName : "+gname+

"\nCity : "+gcity+

"\nBranch : "+gbranch+

"\nTotal : "+gtot+

"\nPercent : "+gper

);

}

else

{

Console.WriteLine("Re-addmission");

}

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Roll : ");

int roll = int.Parse(Console.ReadLine());

Console.Write("Name : ");

string name = Console.ReadLine();

Console.Write("City : ");

string city = Console.ReadLine();

Console.Write("Branch : ");

string branch = Console.ReadLine();

Console.Write("Total : ");

int total = int.Parse(Console.ReadLine());

Console.Write("Percent : ");

int per = int.Parse(Console.ReadLine());

student p1 = new student();

p1.input(roll, name, city, branch, total, per);

Console.ReadLine();

}

}

}

Assignment 2:

Covert any three task function to class and ob**ject using name parameter.**

1. Inheritance
2. Constructor
3. Class (security)
4. Interface
5. Polymorphism
6. Collection class

Inheritance:

* Inheritance is a reusability of class.
* Types of classes in inheritance

1. Base class - All pervious class is called based class(A, B, C)
2. Derived call – after the base class the class is called drive class (B, C, D)

Class A

Class B

Class C

Class D

* Inheritance cannot count main body class.
* In inheritance create only last class object.

How to use inheritance?

With the help of (:) use inheritance property.

Syntax –

Derive class : base class

B:A

D:C

Note: C# cannot support multiple inheritance.

Class A

Class B

Class C : A, B

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace inheritance\_

{

class A

{

public void display()

{

Console.WriteLine("This is first class : ");

}

}

class B : A

{

public void show()

{

Console.WriteLine("This is second class : ");

}

}

class C : B

{

public void print()

{

Console.WriteLine("This is Third class : ");

}

}

class Program

{

static void Main(string[] args)

{

C c1 = new C();

c1.display();

c1.show();

c1.print();

Console.ReadLine();

}

}

}

Assignment 3:

Generate the following result create a class – college

Create a method – input

Accept following parameter college\_id , name, principle, city, state

Create a class – student

Create a method – data input

Parameter is – roll, name, branch, subject, m1, m2, m3, m4 ,m5

Create a class – admission

Create a method – display

Logic 1. Input all parameter by the user

Logic 2. Display all information using display()

Logic 3. Calculate the total

Logic 4. Check the following condition

Total > 250 grade A

Total >. 150 && total < 250 grade B

Total > 100 && total < 150 grade C

Other Wise Fail

All types of parameter is used to name parameter concept.

Class security

1. Abstract class
2. Sealed class

Abstract Class

* Abstract class is use to be created using abstract keyword.
* Abstract class cannot create a object.
* Abstract class only inherit two and classes

Assignment

Create a class – compnay(abstract)

Create a method – salary

Create a member – sal(int)

Value of sal = 5000;

Create a class – employee

Create method – performance ()

Check the following condition

1. Salary > 3000 display A grade employee
2. Salary > 2000 && salary < 3000 display B grade employee
3. Otherwise display new joining.
4. Logic: input salary parameter given by the user.
5. If user cannot input salary then it receive default value
6. Create only last class object

Sealed Class –

* Sealed class access with object
* Sealed class cannot inhert

Convert into sealed class