**C# - Notes**

**15-12-2021**

* **F5 – VS – Call C# Compiler – Build the code – Execute the code – 0/p is displayed on console.**

**Sample Program:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter your name :");

String name = Console.ReadLine();

Console.WriteLine("How much sleep did you have last night");

int hrsOfSleep = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Hello" + name);

if (hrsOfSleep < 8)

{

Console.WriteLine("You must be feeling tired!");

}

else

{

Console.WriteLine("You seem well rested");

}

}

}

}

**Greater or Smaller:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a number :");

int num = Convert.ToInt32(Console.ReadLine());

if (num < 20)

{

Console.WriteLine("{0} is the smallerthan 20", num);

}

else

{

Console.WriteLine("{0} is the greaterthan 20", num);

}

}

}

}

**Else if Program :**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a number :");

int num = Convert.ToInt32(Console.ReadLine());

if (num == 10)

{

Console.WriteLine("The value is 10");

}

else if (num == 20)

{

Console.WriteLine("The value is 20");

}

else if (num == 30)

{

Console.WriteLine("The value is 30");

}

else

{

Console.WriteLine("None of the value");

}

Console.WriteLine("The excat value is {0}", num);

}

}

}

**Odd or Even**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a number :");

int num = Convert.ToInt32(Console.ReadLine());

if (num == 10)

{

Console.WriteLine("The value is 10");

}

else if (num == 20)

{

Console.WriteLine("The value is 20");

}

else if (num == 30)

{

Console.WriteLine("The value is 30");

}

else

{

Console.WriteLine("None of the value");

}

Console.WriteLine("The excat value is {0}", num);

}

}

}

**Negation, AND, OR :**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

int a = 1;

int b = 3;

if (!(a == 1 && b == 2))

{

Console.WriteLine(true);

}

if (a != 1 && b != 2)

{

Console.WriteLine(true);

}

}

}

}

**Create another Class, Constructor :**

**Class main:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Animal animal = new Animal();

animal.MakeNoise();

}

}

}

**Sub class:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

public class Animal

{

public Animal()

{

}

public void MakeNoise()

{

Console.WriteLine("Animal Make Noise");

}

}

}

**Ex**

**Get Employee name from another class:**

**Class1 :**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Employee detail = new Employee();

detail.GetName();

}

}

}

**Class2:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

public class Employee

{

public Employee()

{

}

public void GetName()

{

Console.WriteLine("Enter your name : ");

String name = Console.ReadLine();

Console.WriteLine("Your name is " + name);

}

}

}

**16-12-2021**

**Inheritance:**

**Main:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Emp detail = new Emp();

string E\_name = detail.GetEmpName();

string D\_name = detail.GetDepName();

Console.WriteLine(E\_name + " belongs to " + D\_name);

}

}

}

**Emp class:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

public class Emp : Department

{

public string GetEmpName()

{

return "Venkatesh";

}

}

}

**Department class:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

public class Emp : Department

{

public string GetEmpName()

{

return "Venkatesh";

}

}

}

**Ex**

**Main class:**

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Product purchase = new Product();

Console.WriteLine(purchase.GetOrder() + purchase.GetProdut());

}

}

}

**Order cls:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp2

{

public class Order

{

public string GetOrder()

{

return "Your order is ";

}

}

}

**Product cls:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp2

{

public class Product : Order

{

public string GetProdut()

{

return "Mobile Phone";

}

}

}

**Arthematic Operation:**

**Main cls:**

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

Calculator Calc =new Calculator();

int add = Calc.Add(10, 10);

int sub = Calc.Sub(10, 10);

int mul = Calc.Mul(10, 10);

int div = Calc.Div(10, 10);

Console.WriteLine("\nArthematic Operation");

Console.WriteLine("\n Add = {0}\n Subrcat = {1}\n Multiplication = {2}\n Divide = {3}", add, sub, mul, div);

}

}

}

**Calculator cls:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp3

{

public class Calculator

{

public int Add(int a, int b)

{

return a + b;

}

public int Sub(int a, int b)

{

return a - b;

}

public int Mul(int a, int b)

{

return a \* b;

}

public int Div(int a, int b)

{

return a / b;

}

}

}

**Pass By Value or Pass By References**

using System;

namespace ConsoleApp3

{

class Program

{

static void Inc(int i) //ref int i

{

i = i + 1;

}

static void Main(string[] args)

{

int x = 1;

Inc(x); // ref x

Console.WriteLine(x);

}

}

}

Ans : Using ref key word ans is 2

Without ref key ans is 1

**Interface:**

**Main cls:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

MkSound obj = new MkSound();

obj.AnimalSound();

obj.Birdssound();

}

}

}

**Interface cls:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

interface IBirds

{

void Birdssound();

}

interface IAnimals

{

void AnimalSound();

}

//interface ISound : IBirds, IAnimals { }

class MkSound : IBirds, IAnimals

{

public void Birdssound()

{

Console.WriteLine("Birds Sound");

}

public void AnimalSound()

{

Console.WriteLine("Animal Sound");

}

}

}

**Interface**

**Main cls:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Product obj = new Product();

obj.GetProName();

obj.GetProPrice();

}

}

}

**Interface cls :**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

interface IProduct

{

void GetProName();

void GetProPrice();

}

}

**Sub cls :**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

class Product : IProduct

{

public void GetProName()

{

Console.WriteLine("Product Name = LapTop");

}

public void GetProPrice()

{

Console.WriteLine("Price = 50000");

}

}

}

**Polymorphism:**

**Main:**

using System;

namespace Polymorphism

{

class Program

{

static void Main(string[] args)

{

Animal obj = new Animal();

obj.Greet();

obj.Talk();

obj.Sing();

}

}

}

**Class:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Polymorphism

{

public class Animal

{

public Animal()

{

Console.WriteLine("This is a Constructor");

}

public void Greet()

{

Console.WriteLine("Animal Greet Method");

}

public void Talk()

{

Console.WriteLine("Animal Talk Method");

}

public virtual void Sing()

{

Console.WriteLine("Animal Sang a Song");

}

}

}

**17-12-2021**

**Polymorphism:**

**Virtual -> override for new def same method**

**Main:**

using System;

namespace Polymorphism

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("\n---> Animal Object <---\n");

Animal obj = new Animal();

obj.Greet();

obj.Talk();

obj.Sing();

Console.WriteLine("\n---> Dog Object <---\n");

Animal obj1 = new Dog();

obj1.Greet();

obj1.Talk();

obj1.Sing();

}

}

}

**Animal cls:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Polymorphism

{

public class Animal

{

public Animal()

{

Console.WriteLine("This is a Animal Constructor");

}

public void Greet()

{

Console.WriteLine("Animal Greet Method");

}

public virtual void Talk()

{

Console.WriteLine("Animal virtual Talk Method");

}

public virtual void Sing()

{

Console.WriteLine("Animal Sing virtual Method");

}

}

public class Dog : Animal

{

public Dog()

{

Console.WriteLine("This is a Dog Constructor ");

}

public override void Sing()

{

Console.WriteLine("This is Dog Sing Override Method ");

}

public override void Talk()

{

Console.WriteLine("This is Dog Talk Override Method");

}

}

}

**Bank EX:**

**Main:**

using System;

namespace XYZ\_Bank

{

class Program

{

static void Main(string[] args)

{

Details obj = new Details();

obj.ACC\_Name();

Details obj1 = new Customer();

obj1.ACC\_Name();

}

}

}

**Details :**

using System;

using System.Collections.Generic;

using System.Text;

namespace XYZ\_Bank

{

public class Details

{

public Details()

{

Console.WriteLine("\n---Wellcome to XYZ Bank---\n");

}

public virtual void ACC\_Name()

{

Console.WriteLine("Hii Dear Customer\n");

}

}

public class Customer : Details

{

public Customer()

{

Console.WriteLine("\n--->Customer Details<---\n");

}

public override void ACC\_Name()

{

Console.WriteLine("Enter your name");

String name = Console.ReadLine();

Console.WriteLine("Hii Dear {0}", name);

}

}

}

**Encapsulation:**

**EG:**

**Main:**

using System;

namespace Encapsulation

{

class Program

{

static void Main(string[] args)

{

Account obj = new Account();

double CurrentBalance = obj.GetBalance();

}

}

}

**Sub:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Encapsulation

{

class Account

{

private double balance;

public double GetBalance()

{

return balance;

}

}

}

**Method Overloding:**

**Same method Diff Paramater**

**EG:**

**Main:**

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

TestData obj = new TestData();

Console.WriteLine("{0},{1}",obj.Add(1,2),obj.Add(1,2,3));

}

}

}

**Sub:**

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp2

{

class TestData

{

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

{

return a + b + c;

}

public int Add(int a, int b)

{

return a + b;

}

}

}

**Abstract cls: It have both implement and non-implement method,**

**EG:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Abstract

{

abstract class ABS

{

public int AddTwoNumbers(int a, int b)

{

return a + b;

}

public abstract int SubTwoNumber(int a, int b);

}

}

**Collections:**

**Eg:**

using System;

using System.Collections;

namespace Collection

{

class Program

{

static void Main(string[] args)

{

ArrayList obj = new ArrayList();

obj.Add(12);

obj.Add(12.5);

obj.Add("Two");

int firstelement = (int)obj[0];

double secondelement = (double)obj[1];

string thirdelement = (string)obj[2];

Console.WriteLine("{0},{1},{2}",firstelement, secondelement, thirdelement);

}

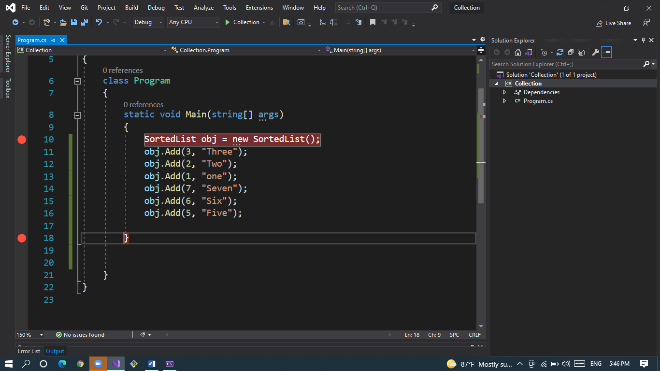
}

}

**Sorted list and debugging**

**F10 is used for debug**

**Eg:**

****

**Stack: eg,**

using System;

using System.Collections;

namespace EG\_stack

{

class Program

{

static void Main(string[] args)

{

Stack mystack = new Stack();

mystack.Push("Hello");

mystack.Push(1);

mystack.Push(2);

mystack.Push(3);

mystack.Push(4);

foreach (var item in mystack)

Console.WriteLine(item);

}

}

}

**Queue: eg;**

using System;

using System.Collections;

namespace EG\_stack

{

class Program

{

static void Main(string[] args)

{

Queue mystack = new Queue();

mystack.Enqueue("Hello");

mystack.Enqueue(1);

mystack.Enqueue(2);

mystack.Enqueue(3);

mystack.Enqueue(4);

foreach (var item in mystack)

Console.WriteLine(item);

}

}

}