using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day7Demos

{

struct Employee

{

int id;

string name;

string dept;

// Employee() { } This will not allow

//Employee(int id, string name) // This is not allowed

//{

// this.id = id;

// this.name = name;

//}

public Employee(int id, string name, string dept)

{

this.id = id;

this.name = name;

this.dept = dept;

}

public void GetDetails()

{

Console.WriteLine("Enter ID");

}

public void DisplayDetails()

{

}

}

//struct PartTimeEmployee : Employee // Inheitance is not there

//{

//}

class StructureDemo

{

static void Main()

{

Employee employee = new Employee();

employee.GetDetails();

employee.DisplayDetails();

}

}

}

using System;

namespace Day7Demos

{

// Data types cud be primitive , UDT > class , struct , enum

struct Dob

{

public int dd, mm, yy;

}

struct Address

{

public string houseNo;

public string StreetName;

string City;

string State;

string PinCode;

public void GetAddressDetails()

{

Console.WriteLine("Enter House No");

houseNo = Console.ReadLine();

Console.WriteLine("Enter Street");

StreetName = Console.ReadLine();

}

}

class Student

{

int rn;

string name;

Dob dob;

Address address;

public void GetDetails()

{

Console.WriteLine("Enter Address");

address.GetAddressDetails();

}

}

class Program

{

static void Main(string[] args)

{

}

}

}

Enum

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day7Demos

{

enum Choice { Addition=1 , Subtraction=3 , Product=9 , Divide=7};

enum Course { ADONet , C, Java}

class EnumDemo

{

public static void Main()

{

Console.WriteLine("Please enter you choice");

int ch = Convert.ToByte(Console.ReadLine());

//if(ch==1)

if(ch== (int)Choice.Addition)

{

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day7Demos

{

class StringVsStringBuilderDemo

{

public static void Main()

{

string str = "";

str += "This";

str += "is";

str += "my";

str += "new";

str += "book";

Console.WriteLine(str);

StringBuilder sb = new StringBuilder();

sb.Append("This");

sb.AppendLine("is");

sb.AppendLine("my");

sb.Append("new");

sb.Append("book");

Console.WriteLine(sb.ToString());

sb.Remove(1, 1);

Console.WriteLine(sb.ToString());

}

}

}

using System;

using System.IO;

namespace Day7Demos

{

// Data types cud be primitive , UDT > class , struct , enum

struct Dob

{

public int dd, mm, yy;

}

struct Address

{

public string houseNo;

public string StreetName;

string City;

string State;

string PinCode;

public void GetAddressDetails()

{

Console.WriteLine("Enter House No");

houseNo = Console.ReadLine();

Console.WriteLine("Enter Street");

StreetName = Console.ReadLine();

}

}

class Student

{

int rn;

string name;

Dob dob;

Address address;

public void GetDetails()

{

Console.WriteLine("Enter Address");

address.GetAddressDetails();

}

}

class Program

{

static void Main(string[] args)

{

//Console.WriteLine("Hello");

//Console.ReadLine();

StringWriter writer = new StringWriter();

writer.WriteLine("This is first line");

writer.WriteLine("This is second line");

StringReader reader = new StringReader(writer.ToString());

string str = reader.ReadLine();

while(str!= null)

{

Console.WriteLine(str);

str= reader.ReadLine();

}

StringReader reader1 = new StringReader(writer.ToString());

char str1 = (char)reader1.Read();

while(str1!=-1)

{

Console.Write(str1);

Console.ReadLine();

str1 = (char)reader1.Read();

}

}

}

}

StreamWriter writer = new StreamWriter("E:\\Practice.txt");

writer.WriteLine("This is my first statement");

writer.WriteLine("This is my second statement");

writer.Close();

StreamReader reader = new StreamReader("E:\\Practice.txt");

//string str = reader.ReadLine();

//while(str!=null)

//{

// Console.WriteLine(str);

// str = reader.ReadLine();

//}

//reader.Close();

Console.WriteLine("using Read Method");

while (reader.Read() > 0)

{

Console.Write((char)reader.Read());

}

// reader.Close();

Console.WriteLine("using Peek Method");

reader = new StreamReader("E:\\Practice.txt");

while (reader.Peek()>0)

{

Console.Write((char)reader.Read());

}

reader.Close();

FileStream fs = new FileStream(@"E:\practiceDemo.txt", FileMode.Create, FileAccess.Write);

StreamWriter writer = new StreamWriter(fs);

Console.WriteLine("Enter Your Name");

string name = Console.ReadLine();

writer.WriteLine(name);

Console.WriteLine("Enter Your Department");

string dept = Console.ReadLine();

writer.WriteLine(dept);

Console.WriteLine("To Read");

writer.Close();

fs.Close();

fs = new FileStream(@"E:\practiceDemo.txt", FileMode.Open, FileAccess.Read);

string str;

StreamReader reader = new StreamReader(fs);

str = reader.ReadLine();

while(str!=null)

{

Console.WriteLine(str);

str = reader.ReadLine();

}

reader.Close();

fs.Close();