using System;

struct record

{

public int rn;

public string name;

public string batch;

//public record() { }

public record(int rn, string name, string batch)

{

this.rn = rn;

this.batch = name = batch;

this.name = name;

}

// NO WE CAN NOT DEFINE DEFAULT CONSTRUCTOR

// THERE CAN BE ONLY ONE FULLY PARAMETERIZED CONSTRUCTOR IN

// A STRUCT

// IT DOES NOT ALLOW INHERITANCE

//public void GetDetails()

//{

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

// Console.WriteLine();

//}

}

class Program

{

static void Main()

{

record rec = new record();

rec.rn = 10;

rec.name = "Ajay";

rec.batch = "B001";

}

}

using System;

struct address

{

public string houseno;

public string streetname;

public string city;

public string state;

public string pincode;

}

struct dob

{

int dd, mm, yy;

}

class Employee

{

int id;

string name;

address add;

dob d\_o\_b;

public void GetDetails()

{

Console.WriteLine("Enter ID");

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

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Address");

Console.WriteLine("Enter House No");

add.houseno = Console.ReadLine();

Console.WriteLine("Enter Street No");

add.streetname = Console.ReadLine();

Console.WriteLine("Enter City");

add.city = Console.ReadLine();

Console.WriteLine("Enter State");

add.state = Console.ReadLine();

Console.WriteLine("Enter Pincode");

add.pincode = Console.ReadLine();

}

}

class Program

{

static void Main()

{

Employee emp = new Employee();

emp.GetDetails();

}

}

Enum > Its is u=sed to assign string constants

represents a group of **constants**

In C#, an enum (or enumeration type) is used to assign constant names to a group of numeric integer values. It makes constant values more readable,

using System;

enum choice {add=1 , subtract , multiply , divide};

class Program

{

static void Main()

{

Console.WriteLine("Enter No1");

int num1 = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter No2");

int num2 = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Choice");

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

switch(ch)

{

case 1: Console.WriteLine(num1+ num2); break;

case 2: Console.WriteLine(num1 - num2); break;

case 3: Console.WriteLine(num1 \* num2); break;

case 4: Console.WriteLine(num1 / num2); break;

default: Console.WriteLine(num1 + num2); break;

}

}

}

using System;

enum choice {add=1 , subtract , multiply , divide};

class Program

{

static void Main()

{

Console.WriteLine("Enter No1");

int num1 = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter No2");

int num2 = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Choice");

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

switch(ch)

{

case (int)choice.add: Console.WriteLine(num1+ num2); break;

case (int)choice.subtract: Console.WriteLine(num1 - num2); break;

case (int)choice.multiply: Console.WriteLine(num1 \* num2); break;

case (int)choice.divide: Console.WriteLine(num1 / num2); break;

default: Console.WriteLine(num1 + num2); break;

}

}

}

<https://www.tutorialsteacher.com/csharp/csharp-enum>