OOPS

1. Class > Class is a user defined type. Blueprint based on which multiple objects of same type could be created
2. Object > Instance or variable of a class

Student > Ajay Deepak Sagar

Data Type > Tells abt the value that can be stored in a variable

Int indicates that only numbers are allowed, we can perform + - \* / on variable of int type

Char only 1 character can be stored

Float

String strcmp strlen strcpy

Bool

Data Types cud be of 2 types

1. Inbuit/primitive , for eg int char float bool string double
2. User Defined Types, for eg class struct enum

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Student

{

int rn;

string name;

string batch;

int marks;

void GetDetails()

{

Console.WriteLine("Enter RollNo");

rn = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Byte.Parse(Console.ReadLine());

}

void DisplayDetails()

{

Console.WriteLine("RollNo is {0}" , rn);

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

Console.WriteLine("Batch is " + batch);

Console.WriteLine("Marks are " + marks);

}

}

class ClassDemo

{

static void Main()

{

// Value type variables take default values when we declare them

int x;

bool b;

// Reference type variables get by defualt null

Student student;

// Right now, this variable/object

// has not been assigned any memory from heap

// It gets allocated memory when we use

// new keyword

student = new Student();

//Student student;

//student = new Student();

Student student1 = new Student();

}

}

}

By default, all class members are private, which means they can not be accessed outside the class. If you want to access them outside the class , we have to use public access specifier

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Student

{

int rn;

string name;

string batch;

int marks;

public void GetDetails()

{

Console.WriteLine("Enter RollNo");

rn = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Byte.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

Console.WriteLine("RollNo is {0}" , rn);

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

Console.WriteLine("Batch is " + batch);

Console.WriteLine("Marks are " + marks);

}

}

class ClassDemo

{

static void Main()

{

// Value type variables take default values when we declare them

int x;

bool b;

// Reference type variables get by defualt null

Student student;

// Right now, this variable/object

// has not been assigned any memory from heap

// It gets allocated memory when we use

// new keyword

student = new Student();

student.GetDetails();

student.DisplayDetails();

//Student student;

//student = new Student();

Student student1 = new Student();

}

}

}

Store & Display records for 2 objects

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Student

{

int rn;

string name;

string batch;

int marks;

public void GetDetails()

{

Console.WriteLine("Enter RollNo");

rn = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Byte.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

Console.WriteLine("RollNo is {0}" , rn);

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

Console.WriteLine("Batch is " + batch);

Console.WriteLine("Marks are " + marks);

}

}

class ClassDemo

{

static void Main()

{

Student student1 = new Student();

student1.GetDetails();

student1.DisplayDetails();

Student student2 = new Student();

student2.GetDetails();

student2.DisplayDetails();

}

}

}

Array of Objects

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SecondProject

{

class Student

{

int rn;

string name;

string batch;

int marks;

public void GetDetails()

{

Console.WriteLine("Enter RollNo");

rn = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Byte.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

Console.WriteLine("RollNo is {0}", rn);

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

Console.WriteLine("Batch is " + batch);

Console.WriteLine("Marks are " + marks);

}

}

class StudentClassDemo

{

static void Main()

{

// Array of Objects

Student[] students = new Student[10];

for (int i = 0; i < 2; i++)

{

Console.WriteLine("Enter Details for Student No {0} ", i + 1);

students[i] = new Student();

students[i].GetDetails();

}

for (int i = 0; i < 2; i++)

{

Console.WriteLine("Details for Student No {0} ", i + 1);

students[i].DisplayDetails();

}

}

}

}

Collection of Students

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ThirdProject

{

class Student

{

int rn;

string name;

string batch;

int marks;

public void GetDetails()

{

Console.WriteLine("Enter RollNo");

rn = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Byte.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

Console.WriteLine("RollNo is {0}", rn);

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

Console.WriteLine("Batch is " + batch);

Console.WriteLine("Marks are " + marks);

}

}

class StudentCollectionClassDemo

{

static void Main()

{

// Collection Objects

ArrayList students = new ArrayList();

for (int i = 0; i < 2; i++)

{

Student student = new Student();

Console.WriteLine("Enter Details for Student");

student.GetDetails();

students.Add(student);

}

foreach(Student student in students)

{

student.DisplayDetails();

}

}

}

}

Generic Collection

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ThirdProject

{

class Student

{

int rn;

string name;

string batch;

int marks;

public void GetDetails()

{

Console.WriteLine("Enter RollNo");

rn = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

marks = Byte.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

Console.WriteLine("RollNo is {0}", rn);

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

Console.WriteLine("Batch is " + batch);

Console.WriteLine("Marks are " + marks);

}

}

class StudentCollectionClassDemo

{

static void Main()

{

// Collection Objects

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

for (int i = 0; i < 2; i++)

{

Student student = new Student();

Console.WriteLine("Enter Details for Student");

student.GetDetails();

students.Add(student);

}

**// students.Add(1);**

foreach(Student student in students)

{

student.DisplayDetails();

}

}

}

}

**Class access specifier is internal, which means it is accessible in the entire project**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class StudentCollectionClassDemo

{

static void Main()

{

// Collection Objects

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

for (int i = 0; i < 2; i++)

{

Student student = new Student();

Console.WriteLine("Enter Details for Student");

student.GetDetails();

students.Add(student);

}

// students.Add(1);

foreach(Student student in students)

{

student.DisplayDetails();

}

}

}

}

Instance Variables

Static Variables

Constant Variable

ReadOnly Variables