Class : user defined type

It’s a template based on which mutiple objects of same type could be created

What is the meaning of Data type ??

It indicates what?

What we can store, which values are allowed

What operations can be done on the values which are allowed with that type

Size (byes to be allocated)

String

Int

All these are pre-defined

Class is a type : user defined

Class student

{

Int rn;

String name;

Void get( ) {}

Void display(){}

}

Int x;

Student s;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Student

{

int rn;

string name;

public void get()

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

}

public void display()

{

Console.WriteLine("RollNo is " + rn);

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

}

}

class Program

{

static void Main(string[] args)

{

int x;

Student student = new Student(); // new will alloctate memory

// it invokes constructor

student.get();

student.display();

}

}

}

---------Procedural programming : Focus is on procedures

Student willcomf enquiry

Counsellor willhandle the enquiry

Stuent qill talke admission

Trainer wil take classess

Studnets will givetest

---- object oriented

Student come for enqi=uiry, tale admission , givetest , come for classes, colect certi

Facutly : give classes , giv test, chektest

Counsellor

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Student

{

int rn;

string name;

string address;

int marks;

public void get()

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public void display()

{

Console.WriteLine("RollNo is " + rn);

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

Console.WriteLine("Address is " + address);

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

}

}

class Program

{

static void Main(string[] args)

{

int x; // = new int();

Student student = new Student(); // new will alloctate memory

// it invokes constructor

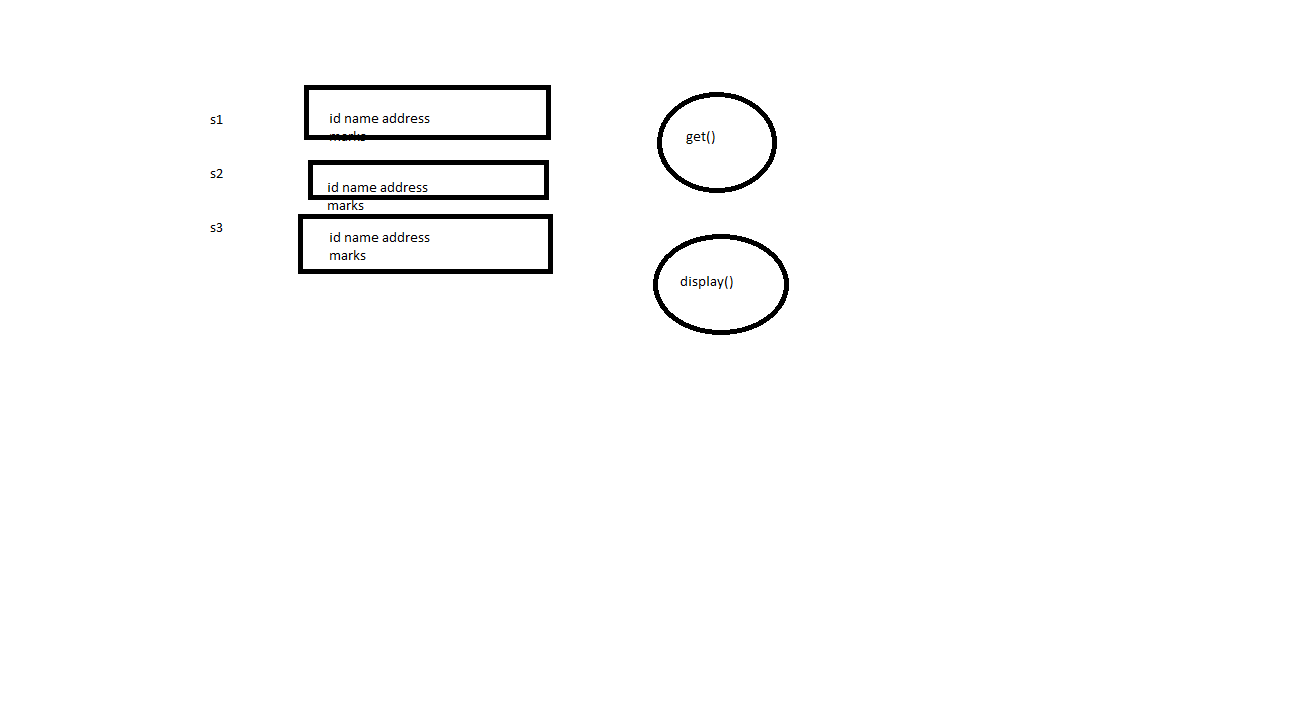
student.get();

student.display();

}

}

}



int x=90;

int x;

Console.Write(“Enter Value”);

Constructors : METhods which are used to initialize values of objects at compilation time

Features of Constructors:

They have same name as class name

They do not have any return type

They are like methods , but we do not call them

They are automatically invoked at time of object declaration

They follow concept of polymorphism

They are never inherited to the child class

**Constructors are of different types**

Default : No parameters, by default, we have a def constructor within a class, but if we create other constructors , then we should define it also, ONLY ONE

Parameterized constructors : where in we provide parameters (THERE COULD BE MULUPLE)

Copy constructors : which are used to copy values of one object to other

Static constructors : which are used to initialize only static variables, They are public by default , but we do not add explictly public, they are parameterless, can be only one in a class

Private constructors : objects could not be create outside the class

using System;

using System.Collections.Generic;

using System.Diagnostics.Contracts;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Student

{

int rn;

string name;

string address;

int marks;

public Student()

{

// Console.WriteLine("Hello");

rn = 10; name = string.Empty; address = string.Empty;marks = 0;

}

public Student(int rn)

{

this.rn = rn;

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name)

{

this.rn = rn;

this.name = name;

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name, string address, int marks)

{

this.rn = rn;

this.name = name;

this.address = address;

this.marks = marks;

}

public void get()

{

Console.WriteLine("Enter RollNo");

this.rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

this.name = Console.ReadLine();

Console.WriteLine("ENter address");

this.address = Console.ReadLine();

Console.WriteLine("Enter Marks");

this.marks = byte.Parse(Console.ReadLine());

}

public void display()

{

Console.WriteLine("RollNo is " + rn);

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

Console.WriteLine("Address is " + address);

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

}

}

class Program

{

static void Main(string[] args)

{

int x; // = new int();

Student student = new Student(); // new will alloctate memory

// it invokes constructor

student.get();

student.display();

Student s1 = new Student(10);

s1.display();

Student s2 = new Student(11, "Ajay", "Delhi", 90);

s2.display();

Student s3 = new Student(name: "Deepak", rn: 20, address: "Delhi", marks: 90);

s3.display();

}

}

}

using System;

using System.Collections.Generic;

using System.Diagnostics.Contracts;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Student

{

int rn;

string name;

string address;

int marks;

public Student()

{

// Console.WriteLine("Hello");

rn = 10; name = string.Empty; address = string.Empty;marks = 0;

}

public Student(int rn)

{

this.rn = rn;

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name)

{

this.rn = rn;

this.name = name;

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name, string address, int marks)

{

this.rn = rn;

this.name = name;

this.address = address;

this.marks = marks;

}

public void get()

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

this.name = Console.ReadLine();

Console.WriteLine("ENter address");

this.address = Console.ReadLine();

Console.WriteLine("Enter Marks");

this.marks = byte.Parse(Console.ReadLine());

}

public void display()

{

Console.WriteLine("RollNo is " + rn);

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

Console.WriteLine("Address is " + address);

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

}

// Copy Constructor

public Student(Student student)

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

name = student.name;

address = student.address;

marks = student.marks;

}

}

class Program

{

static void Main(string[] args)

{

int x; // = new int();

Student student = new Student(); // new will alloctate memory

// it invokes constructor

student.get();

student.display();

Student s1 = new Student(10);

s1.display();

Student s2 = new Student(11, "Ajay", "Delhi", 90);

s2.display();

Student s3 = new Student(name: "Deepak", rn: 20, address: "Delhi", marks: 90);

s3.display();

Student s4 = new Student(s3);

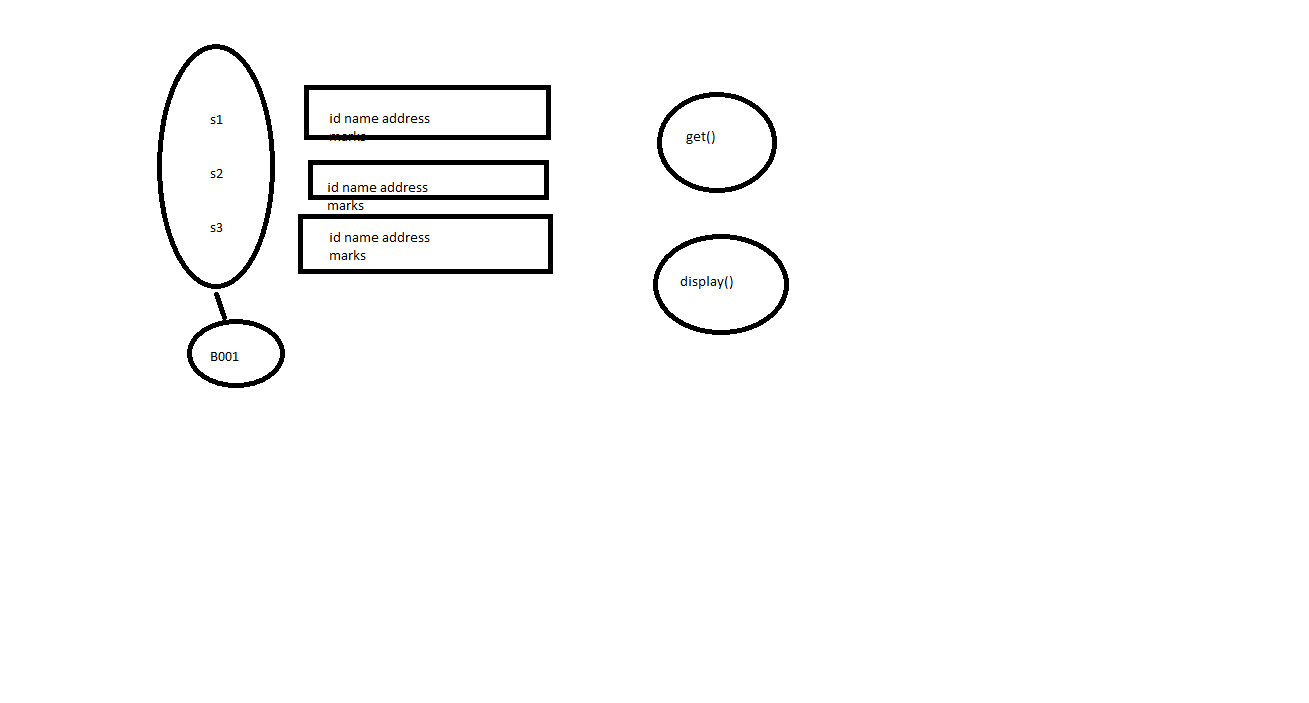
s4.display();

}

}

}

Static variable : a variable which is shared by all the objects. There is a single copy of that variable



Static variables are attached to the class , not to the object

Static Method is a method which can access only static members

using System;

using System.Collections.Generic;

using System.Diagnostics.Contracts;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Student

{

int rn;

string name;

string address;

int marks;

public static string batch="B001";

public Student()

{

// Console.WriteLine("Hello");

rn = 10; name = string.Empty; address = string.Empty;marks = 0;

}

public Student(int rn)

{

this.rn = rn;

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name)

{

this.rn = rn;

this.name = name;

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name, string address, int marks)

{

this.rn = rn;

this.name = name;

this.address = address;

this.marks = marks;

}

public void get()

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

this.name = Console.ReadLine();

Console.WriteLine("ENter address");

this.address = Console.ReadLine();

Console.WriteLine("Enter Marks");

this.marks = byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Batch");

batch= Console.ReadLine();

}

public void display()

{

Console.WriteLine("RollNo is " + rn);

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

Console.WriteLine("Address is " + address);

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

}

// Copy Constructor

public Student(Student student)

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

name = student.name;

address = student.address;

marks = student.marks;

}

public static void GetBatch()

{

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

}

}

class Program

{

static void Main(string[] args)

{

int x; // = new int();

Student.batch = "B002";

Student.GetBatch();

Student student = new Student(); // new will alloctate memory

// it invokes constructor

student.get();

student.display();

Student s1 = new Student(10);

s1.display();

Student s2 = new Student(11, "Ajay", "Delhi", 90);

s2.display();

Student s3 = new Student(name: "Deepak", rn: 20, address: "Delhi", marks: 90);

s3.display();

Student s4 = new Student(s3);

s4.display();

}

}

}

using System;

using System.Collections.Generic;

using System.Diagnostics.Contracts;

using System.Dynamic;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Student

{

int rn;

string name;

string address;

int marks;

public static string batch="B001";

public Student()

{

// Console.WriteLine("Hello");

rn = 10; name = string.Empty; address = string.Empty;marks = 0;

}

public Student(int rn)

{

this.rn = rn;

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name)

{

this.rn = rn;

this.name = name;

Console.WriteLine("ENter address");

address = Console.ReadLine();

Console.WriteLine("Enter Marks");

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

}

public Student(int rn, string name, string address, int marks)

{

this.rn = rn;

this.name = name;

this.address = address;

this.marks = marks;

}

public void get()

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

this.name = Console.ReadLine();

Console.WriteLine("ENter address");

this.address = Console.ReadLine();

Console.WriteLine("Enter Marks");

this.marks = byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Batch");

batch= Console.ReadLine();

}

public void display()

{

Console.WriteLine("RollNo is " + rn);

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

Console.WriteLine("Address is " + address);

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

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

}

// Copy Constructor

public Student(Student student)

{

Console.WriteLine("Enter RollNo");

rn = Convert.ToByte(Console.ReadLine());

name = student.name;

address = student.address;

marks = student.marks;

}

// static Constructors

static Student()

{

batch = "B009";

}

public static void GetBatch()

{

Console.WriteLine("Enter Batch");

batch = Console.ReadLine();

}

}

class Program

{

void get()

{

Console.WriteLine("a");

}

static void get1()

{

}

static void Main(string[] args)

{

Program p = new Program();

p.get();

get1();

int x; // = new int();

Student.batch = "B002";

Student.GetBatch();

Student student = new Student(); // new will alloctate memory

// it invokes constructor

student.get();

student.display();

Student s1 = new Student(10);

s1.display();

Student s2 = new Student(11, "Ajay", "Delhi", 90);

s2.display();

Student s3 = new Student(name: "Deepak", rn: 20, address: "Delhi", marks: 90);

s3.display();

Student s4 = new Student(s3);

s4.display();

}

}

}