Class 7: **class, Properties, Fields Classes 1**

1. Class e always Public Property rakhbo,
2. Private field tokhon e rakhbo jokhon amra kono property customize or logic add korbo.
3. `Fild Should Be Private and Property Should be Public, Logic should be implemented in class not in object: MS recommended`
4. `Person person= New Person(){“AN”, “BN”};`
   `Person person = null , Tar mane tar reference ke Vanish kore dey, kintu object ke vanish kore nah.`
   `Reference Passs hoi Premetive data type e.`

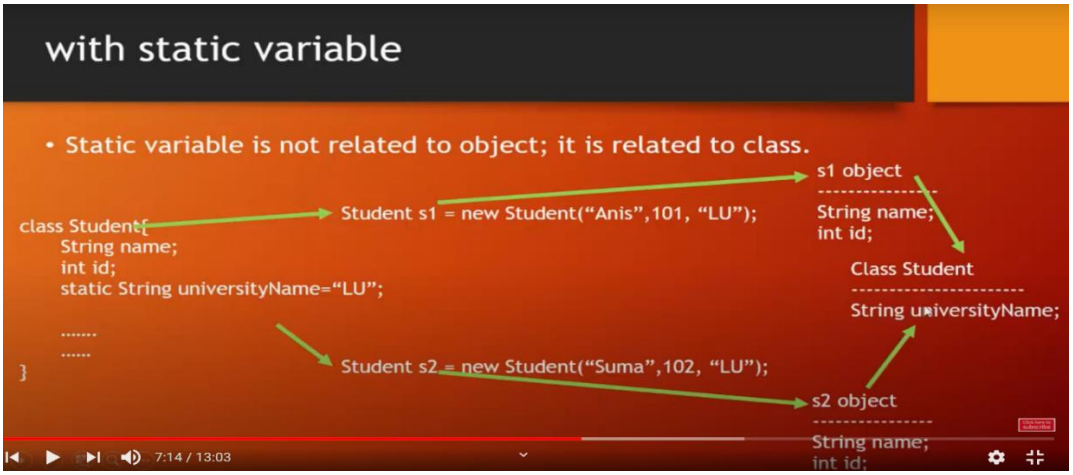
Class 8: **Constructor, Constructor Overloading, Method Overloading, Constructor Chaining, Static class, Stacic Method, Non-Static Method**

1. Constructor: Class er Value jokkhon instance banabo, jeta ekta object hobe, oi object er kichu initial value set korai Constructor er kaj. Student class er Name, address, Date of birth

2. Constructor e Object initialize korar pore, Method updates hoi but field update hoi nah.

   Code: Program.CS : 40

3. Static Method: Tokhon e use korbo jokhon kono Field er Proyojon Hobe nah.



Class 9:**SSH Git**, **Constructor Real Example, Inheritance, All Type of Access Modifiers, Reference adds one project to another project (Class library reference add), Abstract Class**

- Private – Only Own Class // Default variable
- Protected- Own class + Child Class
- Internal – Own Class + Same Project (Onno Project e use kora jabe nah) // Default class
- Protected Internal: Combination
- Public- All Open

2. Class library DLL create kore and console app exe create kore

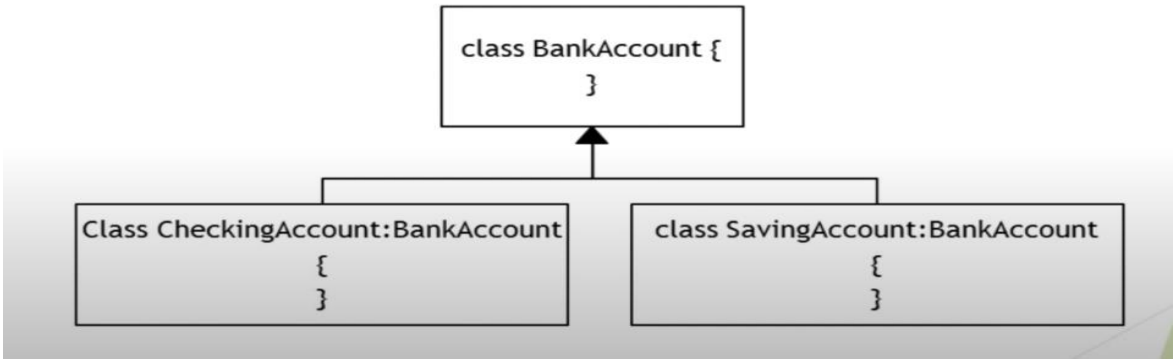
3. Abstract Class: Method Overload korar jonno abstract or virtual use korte hobe at BASE CLASS, and child class override likhte hobe

Extra:

Branch: OOP Example

Project: UpdownCasting ,

Down casting e jawat shomoi Direction dia dite hoi left or right e jabe



5. Child class e object create korar shate shate base class er constructor must call hobey.
6. Override: Run Time, Overload= Compile Time
7. **Summary:**
    - **Virtual Method**- Basic Code thakbe, Pore override kore ni
    - Abstract class e No Obeject, Because Base class e abstract method e body nai. Jodi

- Abstract class e abstract likhe method likhte hobe, Virtual likha jabe nah, pore sheta implement korte hobe

Class 10: **No Class, Exam**

Class 11: **Abstract class Importance, Interface, Multiple Interface, Base Use, Sealed, Partial Class Enum**

1 Abstract class e extra ekta Abstract method declar kora jai, and baki normal class er moto shob kora jai.

2 Why Abstract class? tokhon e use korbo, jokhon kono method enforce korbo child class e, je keo chaile etar object banate parbe na, Plus Kichu common code thakbe , ar jegulo uncertain oigulo child class e korar way kora ase , Design korchi child class e.

3 Virtual je kono class e likhte parbo even property teo, so that jokhon ei class e inherit kore override korte pari.

4 Base, new: Multiple Interface ke base class implement korle, ar oi base class ke onno cornered bottle class jodi base class inherit kore same method thake taile "new" use korbo as a warning minimize er joono, and base "base.") er popr method call dibo.

- **Abstract VS Interface Vs Normal Class**
- **0** Normal class e full body method thakbe, Abstract class e Full body or Incomplete signature method thakte pare, Interface fully incomplete signature method thakbe
- **1** Student: IInformation, IPriter (Ekhane Right side e jeta thakbe oita Base Class
- **2** IInformation reference er moddhe implemented object rakhte pari;
- **3** Fully Abstract in interface, can be fully or partially abstract

Details: https://www.geeksforgeeks.org/difference-between-abstract-class-and-interface-in-c-sharp/

Class 12: **Date Time, String Builder, Params, Ref/In/Out, Tuple, Anonymous Object, Nullable type**

Ref: Value Change hobe, Mandatory na value dewa Body er moddhe.

Out: Value Change hobe, Mandatory value dewa Body er moddhe.

In: `In parameter cannot be changed`

Class 13: **Generic Class, Generic Method, Restriction.**

Restriction: Struct, Class, Interface.

Object Type: Shob type dewa possible.

Extra:

Object Initializer: Assign values to the fields at the time creating an object without invoking CTOR

Collection Initializer: Collection Initializer er Moddhe object initialize hobe.

Code Follow

```
Student student1 = new Student() { Id = 1, Name = "Rasel", Age = 30, Address = "Dhaka" };
Student student2 = new Student() { Id = 2, Name = "Kamrul", Age = 25, Address = "Matlab" };
Student student3 = new Student() { Id = 3, Name = "Arafat", Age = 20, Address = "Chandpur" };
Student student4 = new Student() { Id = 4, Name = "Shakib", Age = 13, Address = "Matlab" };

List<Student> studentList = new List<Student>() {
                        new Student() { Id = 1, Name = "Rasel", Age = 30, Address = "Dhaka" },
                        new Student() { Id = 2, Name = "Kamrul", Age = 25, Address = "Matlab" },
                        new Student() { Id = 3, Name = "Arafat", Age = 20, Address = "Chandpur" },
                        new Student() { Id = 4, Name = "Shakib", Age = 13, Address = "Matlab" }
                                        };
```

One to Many Associations Relationship: Object e null thakle oikhane kono kichu add kora jai na, when ypu debug you can explore. CTOR e list create korar shomoi mus be object create korte hobe. Jekono reference tye er ob create na korle tar method or others functionality access korte parbo na.

Anonymous Type Example: Quickly table theke join kore data ene view model banate hoi, eta na kore amara easily view te dekhate parbo, anonymous type dia transfer kori.