Inheritance

* Pros
* Code Reusability: It allows to use methods and properties of superclass
* Easy to Understand: Helps in modeling real-word relationships
* Polymorphism: Allows objects of derived class to be treated as object of base class
* Cons
* Tight Coupling: Inheritance can lead to tight coupling between classes. Changes in the superclass affect the behaviour of the subclasses
* Inherited Code: Inherit all members of superclass, unnecessary or inappropriate elements for subclasses
* Complexity: As the depth increases it is harder to understand

Polymorphism

* Definition: It means many forms, and it happens when we have many classes that are related to each other by inheritance. Inheritance lets us inherit attributes and methods from another class. **Polymorphism** uses those methods to perform different tasks. This allows us to perform a single action in different ways.
* Static Binding – Compile-Time Polymorphism : It is applied through method overloading
* Dynamic Binding-Run-Time Polymorphism: It is applied through method overriding.

Abstraction

* Definition: It aims to hide complex codes while only exposing the necessary elements.
* By Abstract Classes : These classes have abstract methods and concepts. Properties can not be abstract. Getter and setter methods are abstracted. Can not be instantied on their own
* By Interfaces: These classes has contract of methods that is implemented by concrete classes. Can not be instantied by their own

Encapsulation

* Definition : Encapsulations ties up attributes and methods to single unit known as classes. It could restrict to access to methods and attributes
* Access Modifiers: It is achieved through the use of private, public, protected and internal keywords.

A screenshot of a computer

Description automatically generated

* Properties and Methods : It is applied through use of private and public keywords. A property which is declared private would have getter and setter method. Getter and setter are used to acces and modify the attributes of class.

public long TxnGuid { get; set; }

Association

* Definition: Association between two object when each one of them can use other one, but also each one of them can exist without other one.

A screen shot of a computer program

Description automatically generated

A diagram of composition and composition

Description automatically generated

Composition

* Definition: Composition relationship is formed when a class has a reference to another class as an instance property
* As you can see below, Student class has a relationship with Address. This one to one relationship between classes. Address object can not exist without Student object

A screenshot of a computer code

Description automatically generated

Aggregation

* Definition : Aggregation is another category of "has a" relationship where a class can contain other classes as properties but those classes can exist independently.

