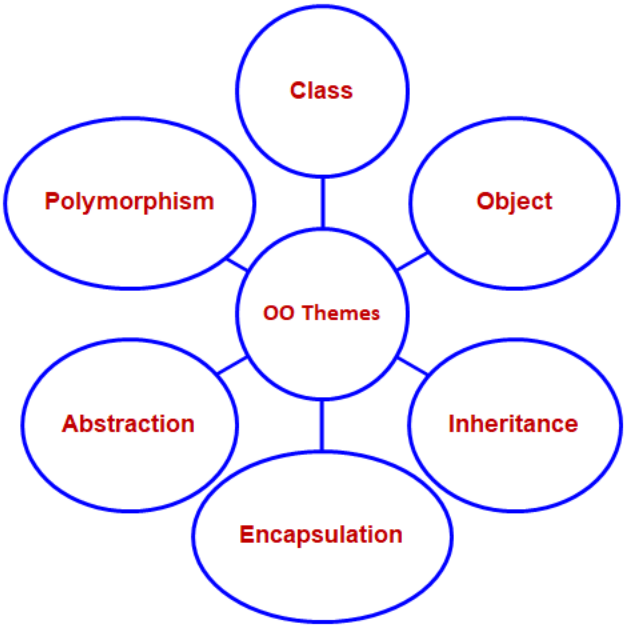
Object Oriented (OO) Themes



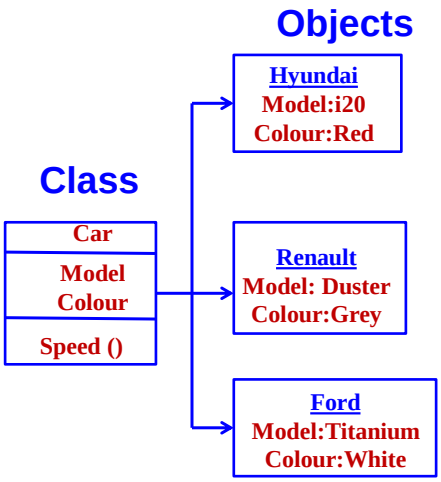
Objects

- In object-oriented analysis, the system can be viewed as a collection of objects
- An *Object* is an identifiable entity with some characteristics and behavior.
- An **object** is an instance of a **class**

Class

- Collection of objects is called *class*. It is a logical entity.
- A *class* is an entity that determines how an object will behave and what the object will contain.
- We can create multiple objects from a class.

Class and Objects

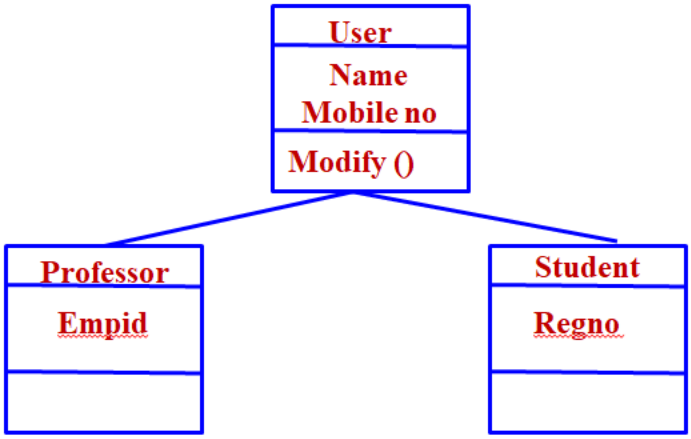


Inheritance

- Objects to be built from other objects
- Relationship between classes(base and derived class)
- OO techniques promote sharing at different levels. Inheritance of both data structure and behavior lets subclasses share common code.

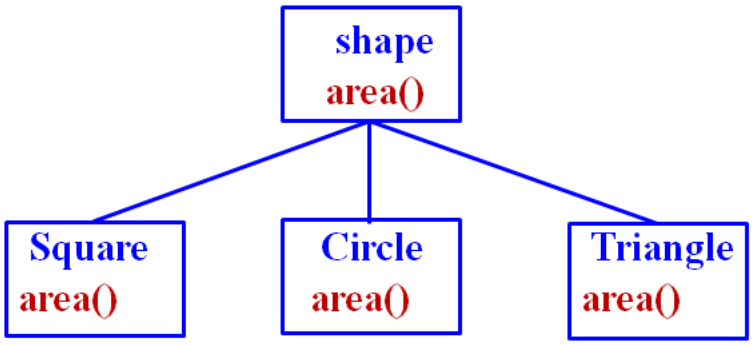
Benefits

- Reduces the number of distinct operations
- Offers reusing designs and code in future projects by investing extra effort in a more general design.



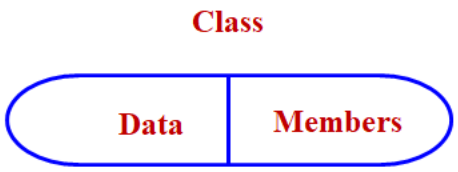
Polymorphism

- Objects that can take many different forms
- Allow us to write generic , reusable code more easily
- Same operation may behave differently on class



Encapsulation and information hiding

- Separates the external aspects of an object that are accessible to other objects, from the internal implementation details that are hidden from other objects.



Abstraction

- Focus on essential aspects of an application while ignoring details. This means focusing on what an object is and does, before deciding how to implement it**.**

Benefits of Object Orientation:

- It becomes easier to understand Complex software systems, as object-oriented structuring provides a closer representation of reality than other programming techniques.
- In a well-made object-oriented system, it should be possible to implement changes at class level, without having to make changes at other points in the system. This reduces the overall amount of maintenance required.
- Object-oriented programming allows you to reuse individual components through polymorphism and inheritance. In an object-oriented system, the amount of work involved in reviewing and maintaining the system is reduced a lot, since many problems can be detected and corrected in the design phase.

Object-oriented Programming Languages: Object-oriented programming techniques do not necessarily depend on object-oriented programming languages. However, the effectiveness of object-oriented programming depends directly on how object-oriented language techniques are implemented in the system kernel of the operating system.

Object-oriented Tools: Object-oriented tools allow you to create object-oriented programs in object-oriented languages. They allow you to model and store development objects and the relationships between them.

Object-oriented Modeling: The object-orientation modeling of a software system is the most important, most time-consuming, and most difficult requirement for attaining the above goals.

Object-oriented design involves more than just object-oriented programming, and it provides various logical advantages that are independent of the actual implementation.

References:

1. Michael R Blaha, James Rumbaugh, Object_oriented Modeling and Design with UML, Second Edition, Pearson Education, 2013