Object Oriented Analysis and Design using Java

UE19CS353

Prof . Sudeepa Roy Dey and Prof. Sindhu R Pai

Department of Computer Science and Engineering

UE19CS353: Object Oriented Analysis and Design using Java

Object Oriented Concepts

Prof . Sudeepa Roy Dey and Prof. Sindhu R Pai

Department of Computer Science and Engineering

Agenda

- ❑ Introduction to OOP

- ❑ Pillars of OOP

- ❑ References
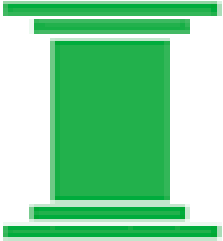
Introduction to OOP

- **Old style programming**:
  - data, which was completely passive
  - functions, which could manipulate any data

- **Object Oriented Programming:**
  - An object contains data and methods that manipulate the data
  - An object is *active,* not passive; it *does* things
  - An object is *responsible* for its own data. It can *expose* the data to other objects

Pillars of OOP

ENCAPSULATION ABSTRACTION INHERITANCE POLYMORPHISM

Abstraction

Encapsulation

Polymorphism

Inheritance

Class
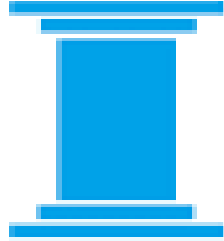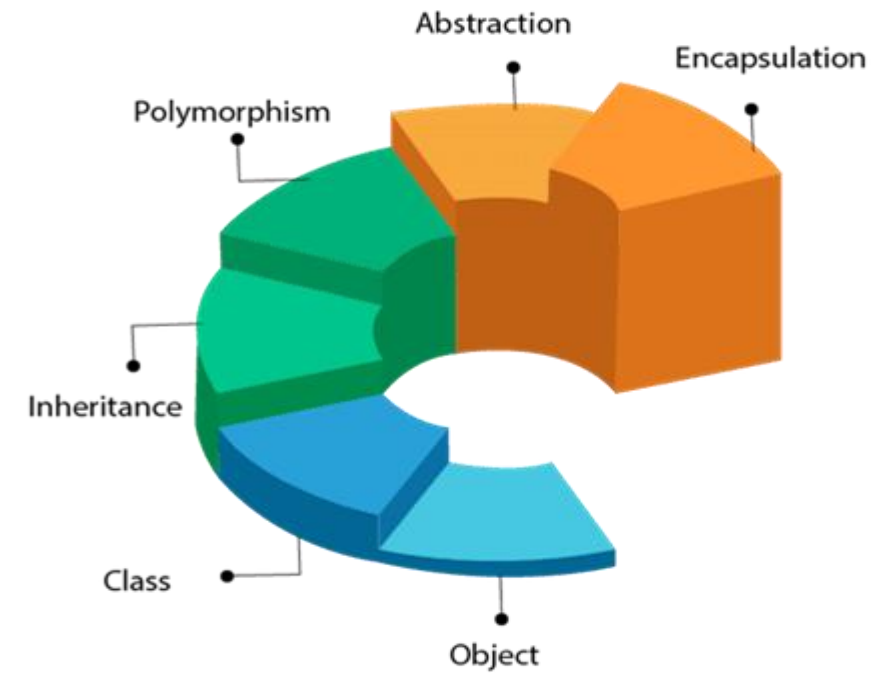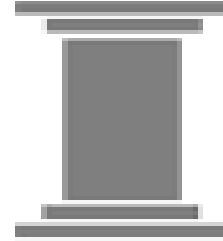
Object

Data Encapsulation

• The wrapping up of data and functions into a single unit is known as encapsulation.

• The data is not accessible to the outside world, only those function which are wrapped in it can access it.

• These functions provide the interface between the object's data and the program.

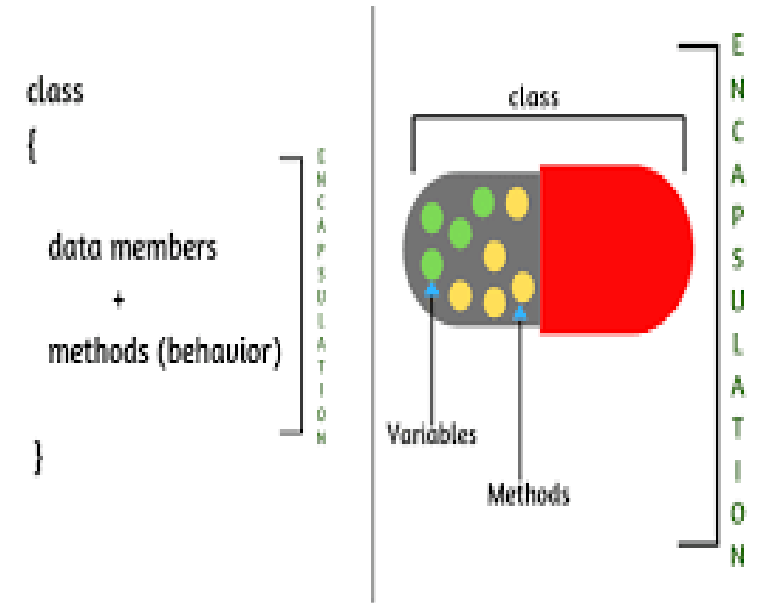• This usage of data from direct access by the program is called data hiding or information hiding.

Fig: Encapsulation

Data Abstraction

- Act of representing essential features without including the background details or explanations.

- Since classes use the concept of data abstraction, they are known as **Abstract Data Types (ADT)**

Polymorphism

- A Greek term means to ability to take more than one form.

- An operation may exhibits different behaviors in different instances. The behavior depends upon the type of data used in the operation.

- The process of making an operator to exhibit different behavior in different instances is known operator overloading
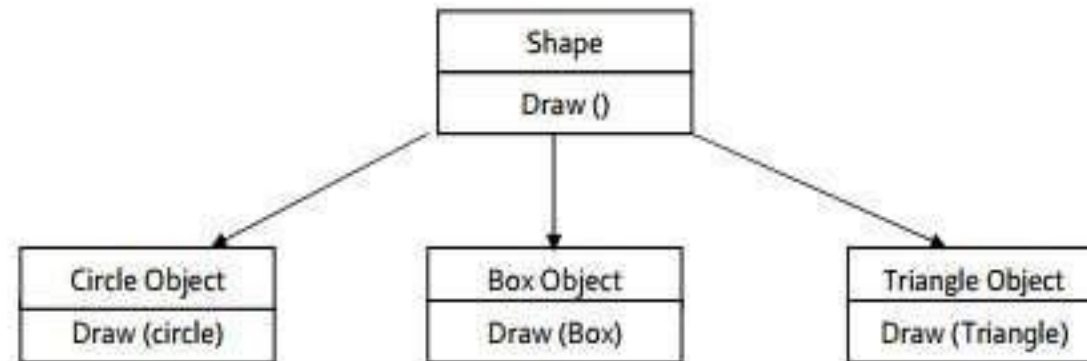
Inheritance

- The process by which objects of one class acquire the properties of objects of another class.

- Provides the idea of reusability. Implies that we can add additional features to an existing class without modifying it.

- Concept of super class/parent class and sub – class/child class
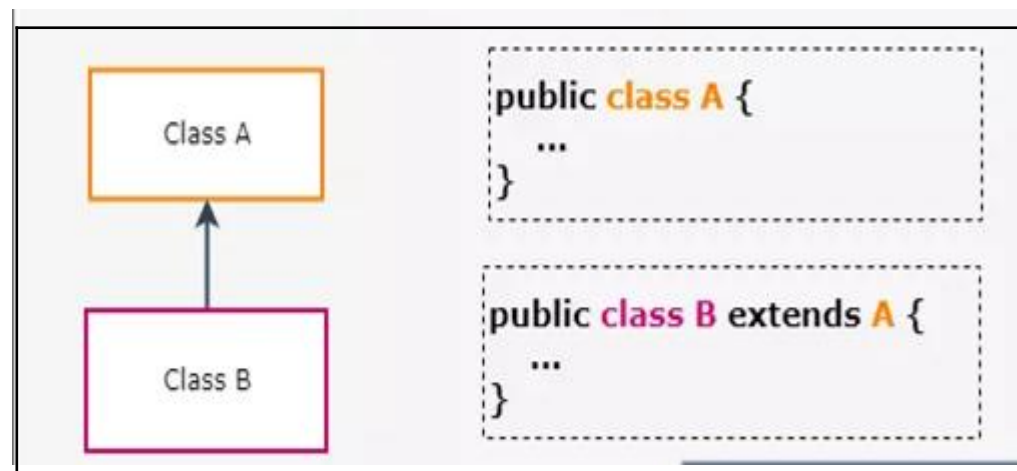
Composition

- To design or implement the "has-a" relationship.

- Composition and Inheritance both are design techniques.

- The Inheritance is used to implement the "is-a" relationship. The "has-a" relationship is used to ensure the code reusability in our program.

- In Composition, we use an instance variable that refers to another object.
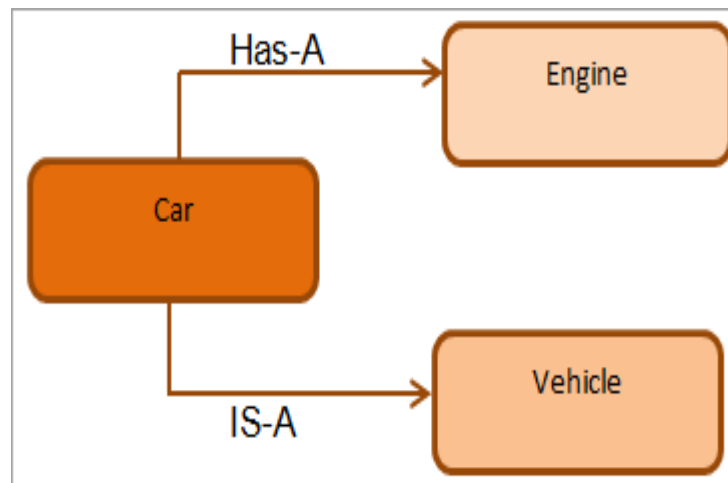
THANK YOU

Sudeepa Roy Dey and Sindhu R Pai

Department of Computer Science and Engineering

sudeepar@pes.edu

sindhurpai@pes.edu

+918277606459