Object Oriented Analysis and Design with Java

UE19CS353

Prof . Sudeepa Roy Dey
Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

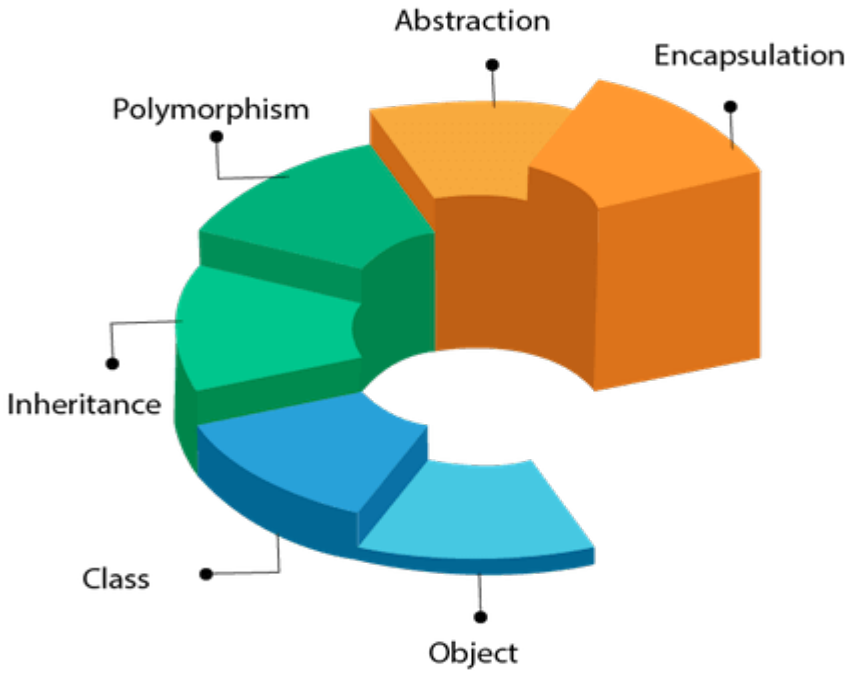
UE19CS353: Object Oriented Analysis and Design with Java

Object Oriented Concepts

Prof Sudeepa Roy Dey
Department of Computer Science and Engineering

Agenda-

- Introduction to object-oriented concepts-Objects and Classes
- Four Pillars of Object Oriented programming
  - Inheritance,
  - Polymorphism
  - Abstraction,
  - Encapsulation



Concept: Introduction to Object Oriented Programming

- In the old style programming, you had:
  - data, which was completely passive
  - functions, which could manipulate any data.

In Object Oriented Programming

- An object contains both data and methods that manipulate that data
  - An object is *active,* not passive; it *does* things
  - An object is *responsible* for its own data
    - But: it can *expose* that data to other objects

Concept: Objects

- An object is a single unit having both data and the processes that operate on that data.
- An object is an entity which has some properties and behavior associated with it.
- Objects are the basic run time entities in an object oriented system.

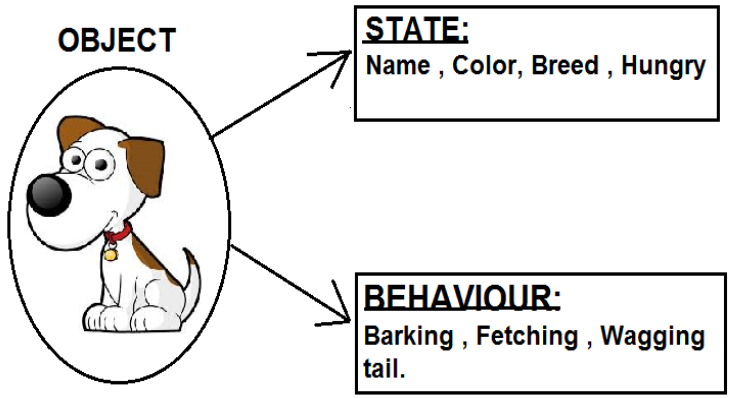
The main purpose of using objects are following.

- They correspond to the real life entities.
- They provide interactions with the real world.
- They provide practical approach for the implementation of the solution.

All the objects have a state, behavior and identity.

Concept: An object has behavior

- An object has Behavior that represents externally visible activities performed by an object in terms of changes in its state.
- Member functions - Member functions represent the code to manipulate the data. The behavior of the object is determined by the member functions



Concept: An object has State

- An object contains both data and methods that manipulate that data
  - The data members represent the state of the object
  - Data can also describe the relationships between this object and other objects

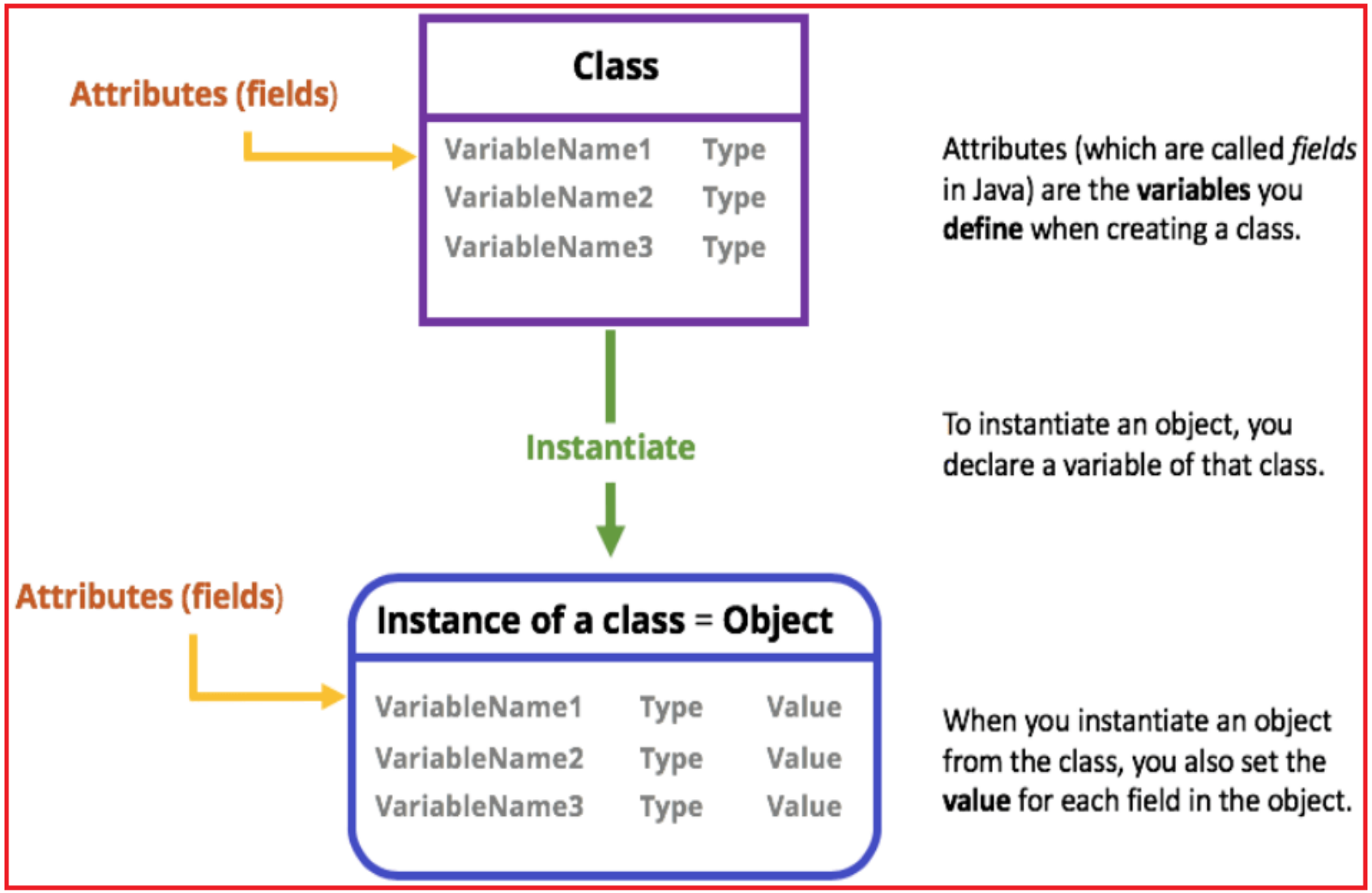
- Example: A CheckingAccount might have
  - A balance (the internal state of the account)
  - An owner (some object representing a person)

Concept: Classes describe objects

- In object-oriented programming, a class is a blueprint for creating objects (a particular data structure)
- Every object belongs to (is an instance of) a class
- An object may have fields, or variables
  - The class describes those fields
- An object may have methods
  - The class describes those methods
- A class is like a template

Object Oriented Analysis and Design with Java

Concept: Classes describe objects



Attributes (which are called *fields* in Java) are the **variables** you **define** when creating a class.

To instantiate an object, you declare a variable of that class.

When you instantiate an object from the class, you also set the **value** for each field in the object.

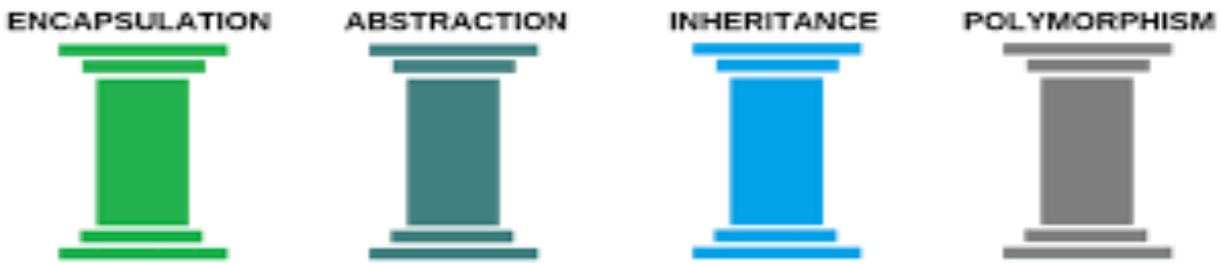
Concept: Classes form a hierarchy

- Classes are arranged in a tree like structure called a hierarchy
- A class, except Object, has a superclass
- A class may have several ancestors.
- When you define a class, you specify its superclass.

  Every class may have one or more subclasses

Concept: Pillars of OOPS

- In this section we will discuss briefly about the four pillars of OOPS.



Concept: 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 can access it.

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

§ This insulation of the data from direct access by the program is called data hiding or information hiding.

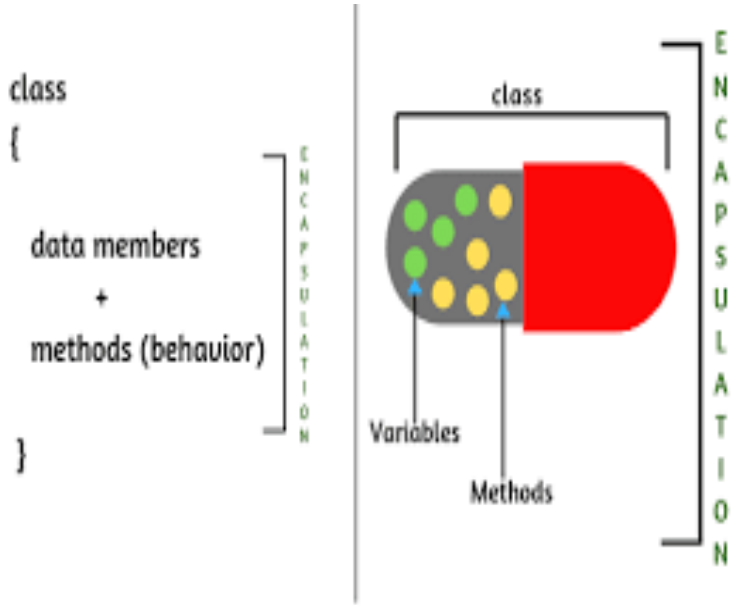
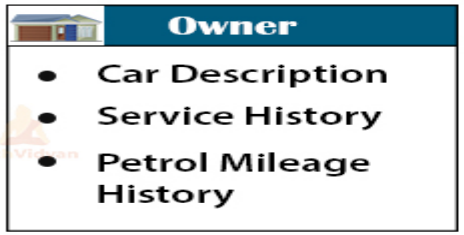


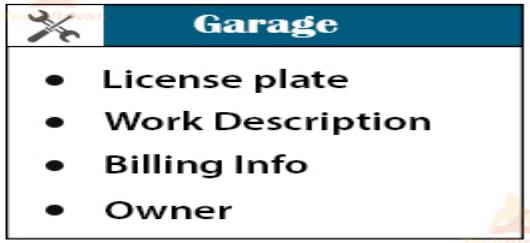
Fig: Encapsulation

Concept: Data Abstraction

§ Abstraction refers to the act of representing essential features without including the background details or explanations.

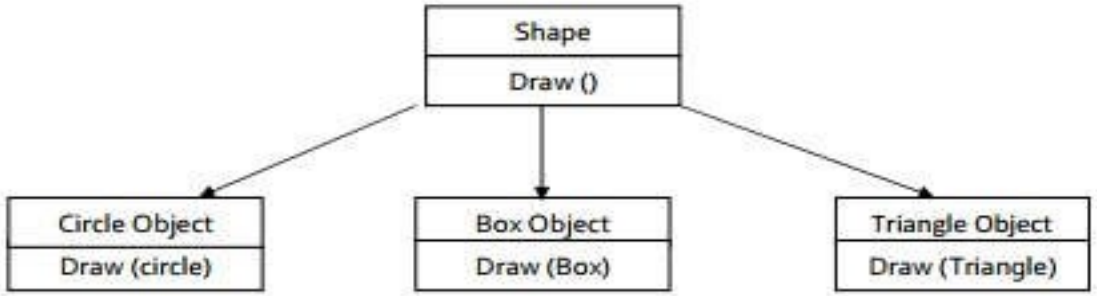
§ Since classes use the concept of data abstraction, they are known as **Abstract Data Types (ADT)**





Concept: Polymorphism

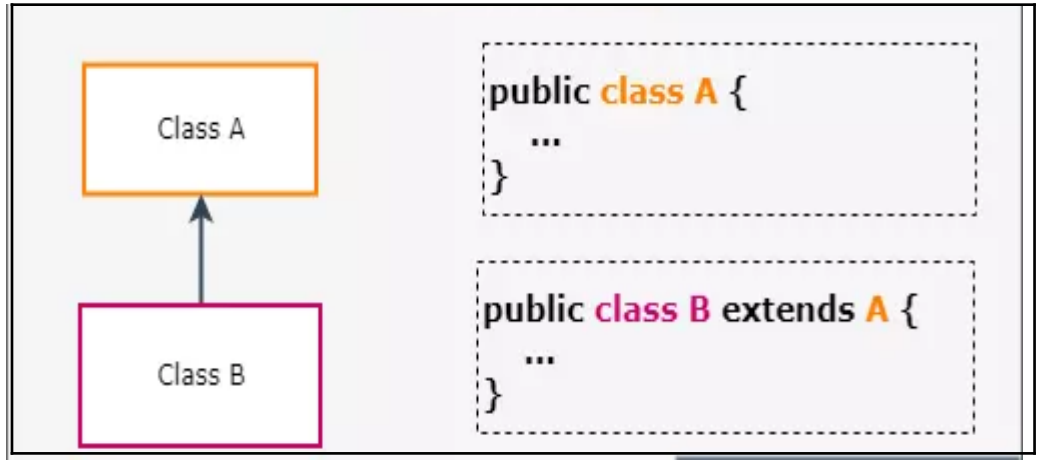
- **Polymorphism,** a Greek term means to ability to take more than one form.
- An operation may exhibits different behaviors in different instances. The behavior depends upon the type of data used in the operation.
- The process of making an operator to exhibit different behavior in different instances is known operator overloading. Java does not support operator overloading though.



Concept: Inheritance

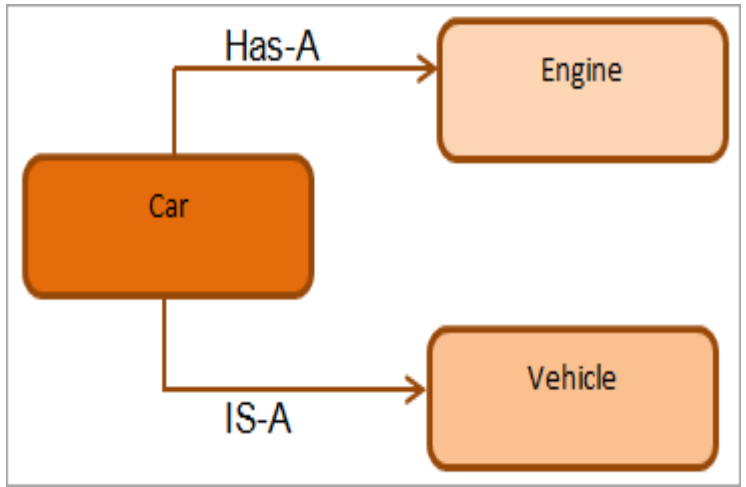
§ **Inheritance** is the process by which objects of one class acquire the properties of objects of another class.

§ In OOP, the concept of inheritance provides the idea of reusability. This means we can add additional features to an existing class without modifying it.



Concept: Composition

- The Composition is a way 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
Department of Computer Science and Engineering

sudeepar@pes.edu