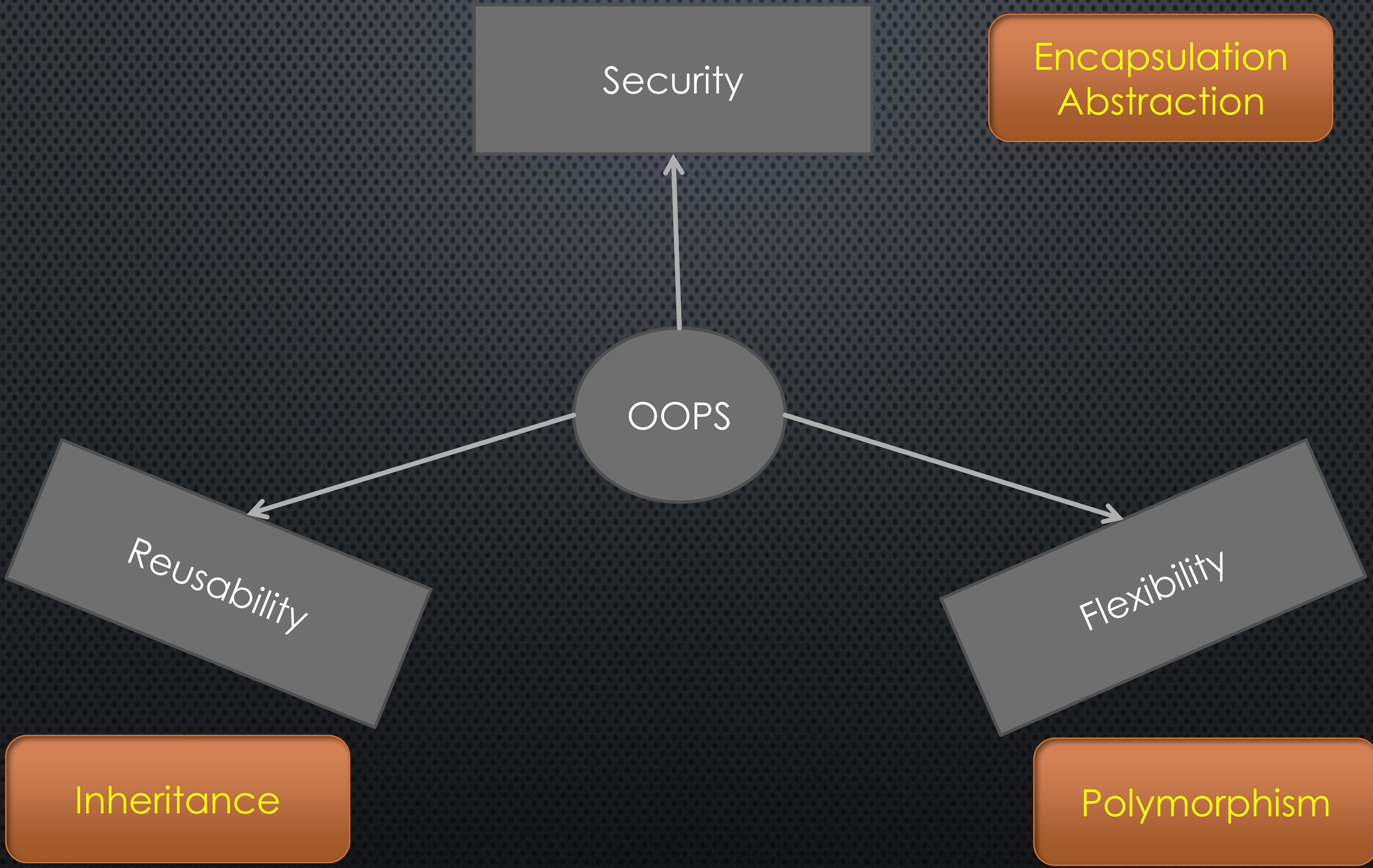
Object Oriented Programming(OOPS)

RAM SHARMA

Pillars of OOPS



DEFINITIONS

- Data Hiding : Hiding of data, so that outside person can’t access our data directly.
  - ❑ By use of ***private***
- Encapsulation: Keeping data and corresponding methods into a single module.
  - ❑ By use of ***class***
- Abstraction: Hiding internal implementation details and showing only functionality to the user.
  - ❑ By use of ***interface and abstract class***
- Inheritance: It allows us to use properties and behavior of one class in another class.
  - ❑ By use of ***extends and implements***
- Polymorphism: It is ability of a method to behave differently in different situation.
  - ❑ By use of ***overloading and overriding***

INHERITANCE

- It is also called as '***Is-a***' relationship.
- The child class will be able to use the states and behavior of its parent class.
- We use ***extends*** keyword to establish parent-child relationship among classes.
- One class can inherit from multiple interfaces but it cannot inherit from multiple other classes. It is called as multiple inheritance. We achieve this using ***implements*** keyword.
- As child inherit from parent, child class reference can call both parent and child class method but parent class reference can call only parent class methods but can't call child specific methods.
- Parent class reference can be used to hold child class objects but vice versa is not possible.
- **Object** class is the parent class of all classes by default.

POLYMORPHISM

- poly means many and morph means forms. many-forms.
- It is achieved in two ways:
  - method overloading or compile-time polymorphism
  - method overriding or run-time polymorphism
- Method name and its argument list is called as method signature.

property	**overloading**	**overriding**
method name	must be same	must be same
signature	must be different	must be same
return type	no restrictions	should be same
private, static, final	supported	not supported
access modifier	no restrictions	can't decrease
resolution	compile-time	run-time

OBJECT CLASS

- Every class has Object as a superclass.
- All the methods of Object class can be called from any reference.
- Important methods :
  - protected Object clone()
  - public boolean equals(Object obj)
  - protected void finalize()
  - public Class<?> getClass()
  - public int hashCode()
  - public String toString()
- hashCode() method gives us address of the object created in heap area.
- default implementation of toString() returns <fully_qualified_classname>@<hashCode()>
- if we try to print any reference, it calls toString() method of its respective class.