OOPs Features – inheritance, interfaces, abstract classes, polymorphism

Inheritance

- To create new classes that are built on existing classes.
- When you inherit from an existing class, you reuse (or inherit) its methods, and you can add new methods and fields to adapt your new class to new situations.

Class, SuperClass, SubClass

- Look for relationships
  - Is student an employee?
  - Is PhD student an student?

- Use extends for inheriting a class

- Superclass/parent/base
- Subclass/child/derived

Quiz

- You have a subclass and a superclass
  - Which typically has more functionality?

Tasks to do

- Create a class and superclass
  - Create private, protected, and public variables and methods in the super class and the subclass
  - Access these variables and methods in subclass and superclass
  - Report your findings

Overriding Methods

- Superclass methods can be redefined in subclasses
  - E.g. salary calculation for different employees may need to be done differently
- How do we distinguish which method id being called
  - User super keyword for calling SuperClass method

- super can also be used to call the constructor of the superclass
  - Must be the first line

Polymorphism

- An object variable (such as the variable e) can refer to multiple actual types

Tasks

- Create at least three classes
  - One superclass, one subclass, one test clss
- Tasks
  - Create a class and superclass
  - Create private, protected, and public variables and methods in the super class and the subclass
  - Access these variables and methods in subclass and superclass
  - Report your findings
  - Create an Array of Superclass object type
    - Initialize with superclass and subclass objects
    - Access methods and see which methods are called

References

- Horstmann, Cay S.. Core Java, Volume I (p. 380). Pearson Education. Kindle Edition.