Lesson - 5 Inheritance - Day - 1 - Topics

What is Inheritance?

Inheritance means deriving a new class from the existing class. Purpose of Reusability.

Inheritance - Generalization (IS-A relationship)

Employee and Manager

To perform inheritance you can use the keyword extends

Class Manger extends from Employee

Inheritance – You can all the fields, methods from the parent except private in your subclass. Like default, public, protected.

Parent Class / Super class

Child Class / Derived class / Sub class

I will create an example to understand the following

1. How to create an Inheritance
2. Super Keyword
3. Rules about parent and child class constructors
4. Overriding(Happen in Parent and Sub class)
5. Polymorphism

Parent class Book , and two sub classes PaperBook and EBook

@Override – Indicates we are overriding the parent method.

This annotation do the compiler verification, incase if change the signature will get error

```
// If you want to acheive polymorphism,
        // Decalre an object as a type of Parent
        // Subclass is a type of Parent too(IS-A relationship)
        Book p1 = new Paparbook();
```

The compiler check the right assignment is the sub type of Left side declaration.

The compile time type of object p1 is Book, and the runtime type of p1 is Paparbook.

Right use of Inheritance must support the following

1. Proper IS-A Relationship
2. Liskov Substitution principle (Whenever you need a parent instance you can substitute subclass instance)

Class A{ }

Class B {}

Class C extends A, B // Not allowed in Java – Multiple Inheritance

To overcome the multiple inheritance issue, Java support Multiple interface implementation allowed

Integer, Float, Double, Long, Byte → inherits from Number

generalization

StaffPerson person1 = new Professor():

StaffPerson person2 = new Secretary();

This is similar in spirit to the automatic conversions that are done for primitive types:

byte b = 8;

int k = b;

Manager subclass of Employee

super Keyword

@Override

Polymorphic Type and Dynammic binding

Rules for Overriding

Access Modifiers

LSP (correct use of Inheritance)

Rules about subclass Constructor

instanceof

Order of Execution