Agenda

- Association
- Inheritance
- super keyword
- Types of inheritance
- Method Overriding
- ~~Upcasting & Downcasting~~
- ~~Final Method & Class~~

Inheritance

- If "is-a"/"kind-of" relationship exist between the types, then use inheritance.
- Inheritance is process of generalization to specialization.
- All members of parent class are inherited to the child class.
- Parent class is also called as super class and child class is also called as sub-class.
- Example:
  - Manager is a Employee
  - Mango is a Fruit
  - Rectangle is a Shape
- In Java, inheritance is done using extends keyword.
- Java doesn't support multiple implementation inheritance i.e. a class cannot be inherited from multiple super-classes.
- However Java does support multiple interface inheritance i.e. a class can be inherited from multiple super interfaces.

Super Keyword

- In sub-class, super-class members are referred using "super" keyword.
- used for calling super class constructor
- By default, when sub-class object is created, first super-class constructor (param-less) is executed and then sub-class constructor is executed.
- "super" keyword is used to explicitly call super-class constructor.
- Super class members (non-private) are accessible in sub-class directly or using "this" reference. These members can also be accessed using "super" keyword.
- However, if sub-class method signature is same as super-class signature, it hides/shadows method of the super class i.e. super-class method is not directly visible in sub-class.
- The "super" keyword is mandetory for accessing such hidden members of the super-class.

Types of Inheritance

- 1. Single

```
class A {

}
class B extends A{
}
```

- 2. Multiple

```
class A {

}
class B {

}
class C extends A,B{ // Not Allowed

}

interface I1{

}
interface I2{

}

interface I3 extends I1,I2{ // Allowed

}

class D implements I1,I2{ // Allowed

}
```

- 3. Hirerachical

```
class A {

}
class B extends A{

}
class C extends A{

}
```

- 4. Multilevel

```
class A {

}
```

```
class B extends A{

}
class C extends B{

}
```

- Hybrid inheritance: Any combination of above types

Method Overriding

- Redefining a super-class method in sub-class with exactly same signature is called as "Method overriding".
- Programmer should override a method in sub-class in one of the following scenarios
    - 1. Super-class has not provided method implementation at all (abstract method).
    - 2. Super-class has provided partial method implementation and sub-class needs additional code. Here sub-class implementation may call super-class method (using super keyword).
    - 3. Sub-class needs different implementation than that of super-class method implementation.