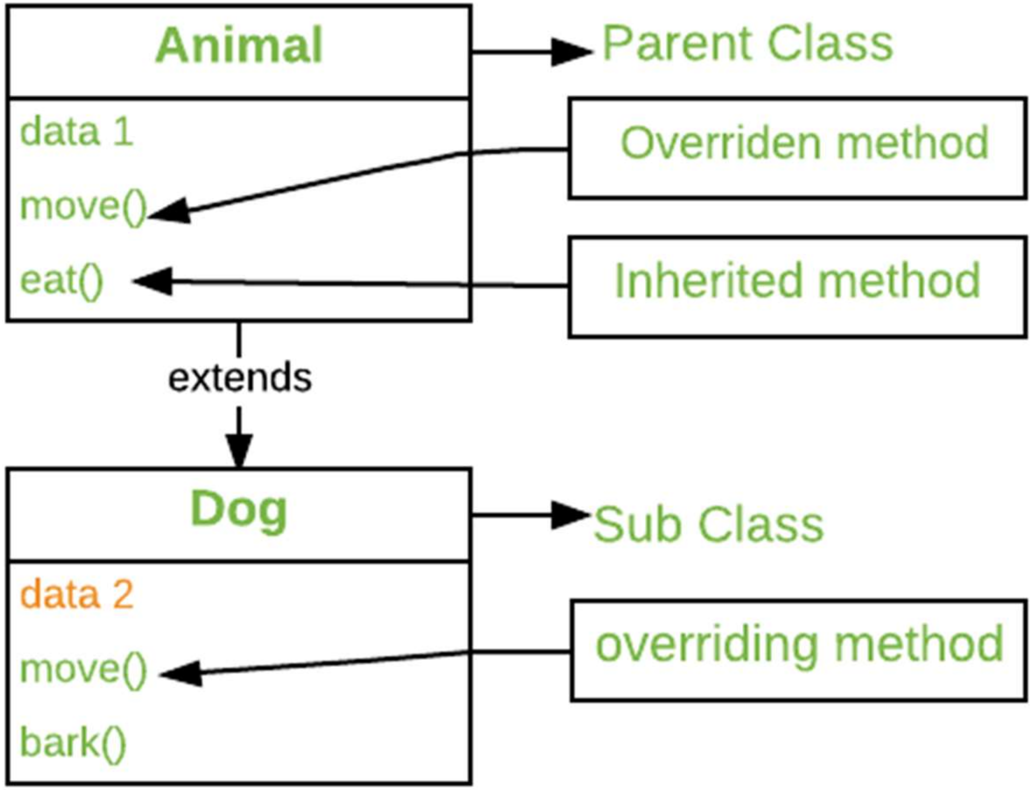
#17

OVERRIDING

Overriding

- If a class inherits a method from its superclass, then there is a chance to override the method provided that it is not marked final.
- In object-oriented terms, overriding means to override the functionality of an existing method.



ZOOMING

Overriding - Example

```java
class Parent {
    void show()
    {
        System.out.println("Parent");
    }
}
// Inherited class
class Child extends Parent {
    void show()
    {
        System.out.println("Child");
    }
}
```

```java
// Driver class
class Main {
    public static void main(String[] args)
    {
        Parent obj1 = new Parent();
        obj1.show();
        Parent obj2 = new Child();
        obj2.show();
    }
}
```

Overriding - Example

```
class Parent {
    void show()
    {
        System.out.println("Parent");
    }
}
// Inherited class
class Child extends Parent {
    void disp()
    {
        System.out.println("Child");
    }
}
```

```
// Driver class
class Main {
    public static void main(String[] args)
    {
        Parent obj1 = new Child();
        obj1.disp();

        Child obj2 = new Child();
        obj2.disp();

    }
}
```

```
Error.
Because, obj1 cant access disp
```

Overriding - Example

```java
class Parent {
    final void show()
    {
        System.out.println("Parent");
    }
}
// Inherited class
class Child extends Parent {
    void show()
    {
        System.out.println("Child");
    }
}
```

```java
// Driver class
class Main {
    public static void main(String[] args)
    {
        Parent obj1 = new Parent();
        obj1.show();

    }
}
```

```
Error.
Final cant override
```