URBAN EDGE

Chapter 11 - Abstract Classes & Interfaces

What does Abstract (class) mean?

Abstract in english means → existing in thought or as an idea without concrete existance

Abstract method

A method that is declared without an implementation

```
abstract void moveTo (double x, double y)
```

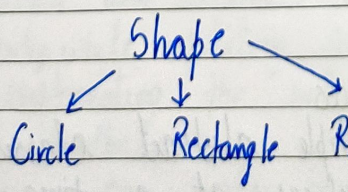
Abstract Class

If a class includes abstract methods, then the class itself must be declared abstract, as in:

```
public abstract class PhoneModel {
    abstract void switchOff();
    // more code
    }
```

When an abstract class is subclassed, the subclass usually provides implementations for all of the methods in parent class. If it doesnt, it must be declared abstract

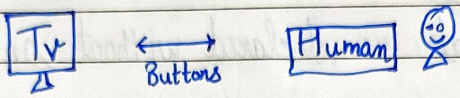
An Example



Note - It is possible to create reference of an abstract class
It is not possible to create an object of an abstract class

We can also assign reference of an abstract class to the object of a concrete subclass.

Interfaces in Java
Interface in english is a point where two systems meet and interact



In Java interface is a group of related methods with empty bodies

An Example

```
interface Bicycle {
    void applyBrake(int decrement);
    void speedUp(int increment);
    }

class AvonCycle implements Bicycle {
    int speed = 7;
    void applyBrake(int decrement) {
        speed = speed - decrement;
        }
    void speedUp(int increment) {
        speed = speed + increment;
        }
```

Abstract class vs Interfaces
We cant extend multiple abstract classes but we can implement multiple interfaces at a time.
Interfaces are meant for dynamic method dispatch

and run time polymorphism

Is multiple inheritance allowed in Java?
Multiple inheritance face problems when there exist methods with same signature in both the super classes.
Due to such problems, Java does not support multiple inheritance directly but the similar concept can be acheived using Interfaces
A class can implement multiple Interfaces and extend a class at the same time.

Note: ① Interfaces in Java is a bit like the Class but with a significant difference.
② An Interface can only have method signatures, constant ← fields and default methods.
③ The class implementing an Interface needs to ✓defined ~~declare~~ the methods (~~not fields~~) ✓not necesaryly fields
④ You can create a reference of Interfaces but not the Object
⑤ Interface methods are public by default.

Default methods
An interface can have static and default methods.
Default methods enable us to add new functionality to existing Interfaces.
This feature was introduced in Java 8 to ensure backward compatibity while updating an Interface.
Classes implementing the interface need not implement the default methods.
Interfaces can also include private methods for default methods to use.

URBAN EDGE

Inheritance in Interfaces

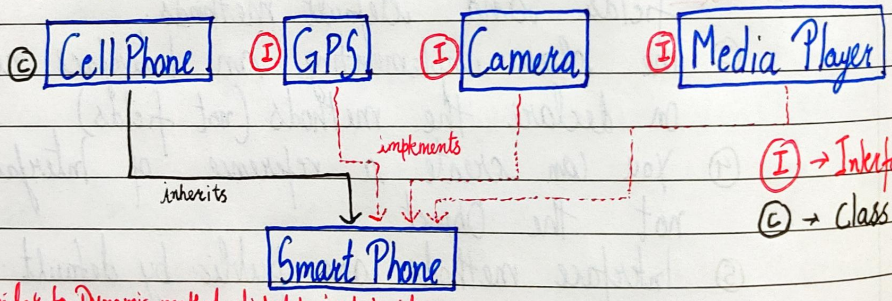
Interfaces can extend another interfaces:

```
public interface Interface1 {
        Void meth1();
    }

public interface Interface2 extends Interface1 {
        Void meth2();
    }
```

Remember that interface cannot implement another interface, only classes can do that!

Polymorphism using Interfaces



Similar to Dynamic method dispatch in Inheritance

- `GPS g = new SmartPhone();` → Can only use GPS methods
- `SmartPhone s = new SmartPhone();` → Can only use SmartPhone methods

Implementing an Interface forces method implementation.