

Lesosn-5-Abstract class and Interfaces

Abstract class – Generalization, what to do and not how to do.

- Inheritance rules are same for Abstract class
- If you want to make your class as abstract use the keyword
- You can make your class as abstract even if you don't have abstract methods.
- Abstract class can have abstract methods, instance methods and instance fields, constructors
- Constructor should not be private
- If it is a abstract method, you should not provide body and declare the method as abstract.
- Don't abstract method as static
- If your class is abstract you cannot instantiate, you can an object of abstract class.

In UML diagram the abstract and interface can use with Stereo type << >>

<<abstract>> , <<interface>>

Interfaces

- ➔ One class can implement multiple interfaces
- ➔ You cannot create an object for the Interface
- ➔ Interface does not have constructors
- ➔ Interface can have fields are public static final and methods are public abstract. (Pre-Java-8)
- ➔ Implement the abstract method inside the class

Example

```
class c1 implements intFace1 {}
```

```
class c1 implements intFace1, intFace2 { }
```

Interfaces (Pre-Java 8 and Java 8) – Refer Slide – 29 – To know the differences

FlashDrive

HP, Sony , Scandisk

Functionality

Read files

Store files

Deletefiles

Modify

Interface in Java- 8 – Clash Rules

Rule 1 : Interface vs Interface – Two interface has the same method with the same signature, if both are unimplemented no issue or no clash. But if both methods are implemented or any one method is implemented using default keyword, you will get a clash. To overcome this error you declare the method in an interface as default or abstract

If two interface has the implemented methods, If your interface wants the specific implementation by calling `Interfacename.super.methodname`.

Rule 2: Interface vs super Class – No clash, your superclass automatically wins.