Agenda

- Revsion
- Abstract class/method
- Interfaces
- Marker interfaces

Abstract Method/class (Demo01 & Demo02)

- If the implementation of any method in superclass is 100% incomplete then such methods should be declared as abstract
- abstract method do not have any method body. these method declaration are only kept inside the class.
- the class in which abstract methods exist, such classes are called as abstract classes.
- In java, we have to declare such classes as abstract using the abstract access modifier
- abstract methods must be implemented by the sub classes.
- if they are not implemented into the subclass then the subclass also need to be declaed as abstract.
- we cannot create object of the abstract class, we can only create the reference.
- abstract classes can have abstract as well as non abstract methods.
- abstract classes can have static as well as non static fields and constructor as well

Interface (Java 7 Interface) (Demo03 & Demo04)

- It provide set of rules/protocols.
- It is the link between service provider and service consumner
- Interfaces can be used for related as well as unrelated classes.
- to create an interface we have to use the keyword `interface`

```
public interface Accecptable{
    void accept();
}
```

- interface consists of only abstract methods.
- all the methods declared inside the interface are by default public and abstract
- class implements the interface
- classes that implements the interface have to compulsary override all the abstract methods from the interface into their class.
- multiple inheritance is supported in interface

```
interface I1{

}
interface I2 {

}
interface I3 extends I1,I2 // Allowed
```

```
{

}
```

Marker interface (Demo05)

- An interface wit no any methods.
- An empty interface is called as marker interface
- It is used to provide an extra information to the JVM about the class that have implemented this interface
- eg
    - java.lang.Cloneable
    - java.io.Serializable

Cloneable Interface Demo

- We cannot cretae object copies directly by assigning it ro references.
- it generally copies the reference, where both references points at same object.
- to crete copy of the object we have to override object class clone method.
- object class clone method does shallow copy.
- It throws an exception CloneNotSupportedException if your class does not implements the Cloneable Marker interface

Lab Work

- solve assignemnts
- go through slides and demmo if required
- class,inheritance,upcasting and exception handling revision