INTERFACES & INHERITANCE

INTERFACES IN JAVA

- In the Java programming, an interface is a reference type, similar to a class, that can contain only constants, method signatures, default methods, static methods and nested interfaces.

DEFINING AN INTERFACE

- An interface declaration consists of modifiers, the keyword interface, the interface name, a comma-separated list of parent interfaces (if any), and the interface body.

```
interface MyInterface {
    public void method1();
    public void method2();
}
```

IMPLEMENTING AN INTERFACE

- To declare a class that implements an interface, you include an implements clause in the class declaration. Your class can implement more than one interface, so the implements keyword is followed by a comma-separated list of the interfaces implemented by the class. By convention, the implements clause follows the extends clause, if there is one.

IMPLEMENTING AN INTERFACE

```
class XYZ implements MyInterface{
    public void method1(){
        System.out.println("implementation of method1");
    }
    public void method2(){
        System.out.println("implementation of method2");
    }
    public static void main(String arg[]){
        MyInterface obj = new XYZ();
        obj. method1();
    }
}
```

INHERITANCE (1)

- Classes can be derived from other classes, thereby inheriting fields and methods from those classes.
- A class that is derived from another class is called a subclass.
- The class from which the subclass is derived is called a superclass.

INHERITANCE (2)

- The idea of inheritance is simple but powerful: reuse.
- A subclass inherits all the members (fields, methods, and nested classes) from its superclass. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.

```
public class ChildClass extends BaseClass{
    …
}
```

METHOD OVERRIDING

- Declaring a method in subclass which is already present in parent class is known as method overriding.
- The main advantage of method overriding is that the class can give its own specific implementation to a inherited method without even modifying the superclass.

ACCESSING SUPERCLASS MEMBERS

- If your method overrides one of its superclass's methods, you can invoke the ovveridden method through the use of the keyword super. You can also use super to refer to a hidden field.
- Invocation of a superclass constructor must be the first line in the subclass constructor.

ABSTRACT METHODS AND CLASSES

- An abstract class is a class that is declared **abstract** – it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed.
- An abstract method is a method that is declared without an implementation.

ABSTRACT CLASSES VS. INTERFACES

- Abstract classes are similar to interfaces. You cannot instantiate them, and they may contain a mix of methods declared with or without an implementation.
- However, with abstract classes, you can declare fields that are not static and final, and define public, protected and private concrete methods.