Abstraction Techniques

1 Abstract classes and methods

- Abstract methods have abstract in the signature
- Abstract methods have no body
- Abstract methods make the class abstract
- Abstract classes can't be instantiated
- Concrete subclasses complete the implementation

2 Multiple inheritance

- Having a class inherit directly from multiple ancestors
- Each language has its own rules
- Java forbids it for classes
- Java permits it for interfaces - no competing implementation

3 Interfaces as types

- Implementing classes don't inherit code
- However implementing classes are subtypes of the interface type
- So, polymorphism is available with interfaces as well as classes

4 Features of interfaces

- All methods are abstract
- There are no constructors
- All methods are public
- All fields are public, static and final

5 Interfaces as specifications

- Strong separation of functionality form implementation - though parameter and return types are mandated
- Clients interact independently of the implementation - but clients can choose from alternative implementations

6 Example of an interface

```
public interface Actor
{
        /**
        * Perform the actor's regular behavior.
        * @param newActors A list for storing newly created
        *                  actors.
        */
        void act(List<Actor> newActors);

        /**
        * Is the actor still active?
        * @return true if still active, false if not.
        */
        boolean isActive();
}
```