Interfaces, Abstract Classes and the DanceStudio

- Similarities and Differences between Abstact Classes and interfaces

- Dance Studio Project

Class Object

- In Java every class by default extends a library class Object (from java.lang)
- Object is a concrete class

```
public class Object
{
    public String toString {...}
    public boolean equals (Object other) {... }
    public int hashCode() { ... }

    // a few other methods
    ...
}
```

Methods redefined (overridden) as necessary

Similarities

Classes	Interfaces
• A superclass provides a secondary data type to objects of its subclasses.	• An interface provides a secondary data type to objects of classes that implement that interface.
• An abstract class cannot be instantiated.	• An interface cannot be instantiated.

Similarities

Classes	Interfaces
• A concrete subclass of an abstract class must define all the inherited abstract methods.	• A concrete class that implements an interface must define all the methods specified by the interface.
• A class can extend another class. A subclass can add methods and override some of its superclass' methods.	• An interface can extend another interface (called its *superinterface*) by adding declarations of abstract methods.

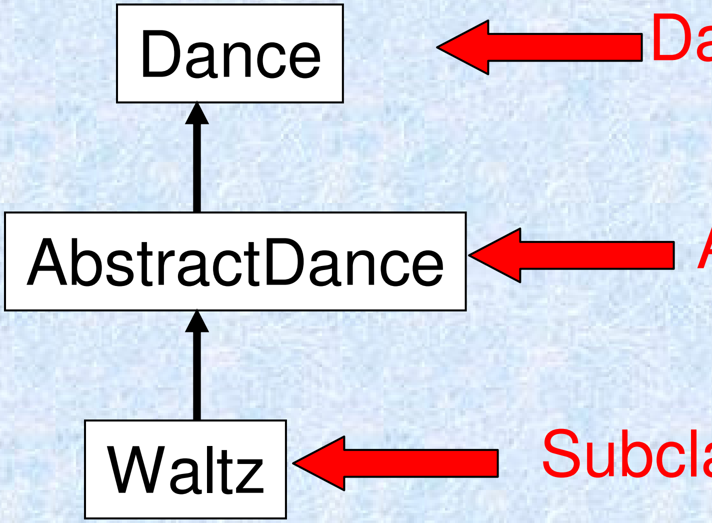
Differences

Classes	Interfaces
• A class can extend only one class.	• A class can implement any number of interfaces.
• A class can have fields.	• An interface cannot have fields (except, possibly, some public static final constants).
• A class defines its own constructors (or gets a default constructor).	• An interface has no constructors.

Differences

Classes	Interfaces
• A concrete class has all its methods defined. An abstract class usually has one or more abstract methods.	• All methods declared in an interface are abstract.
• Every class is a part of a hierarchy of classes with Object at the top.	• Interfaces are generally standalone structures.

Dance Studio (already done)

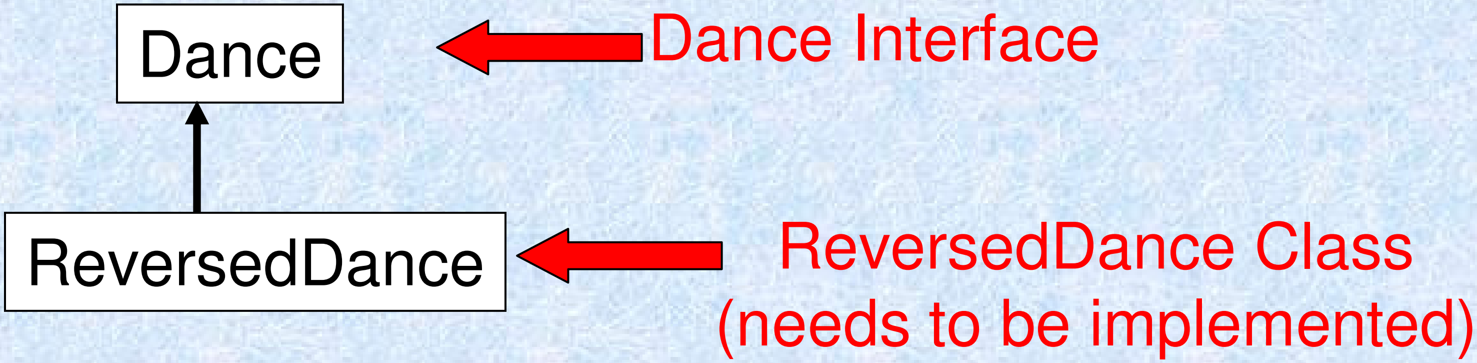


Dance Interface

AbstractDance Class

Subclass of AbstractDance

Dance Studio (your job)



Dance

ReversedDance

Dance Interface

ReversedDance Class
(needs to be implemented)

```
public interface Dance
{
    DanceStep getStep(int i);
    int getTempo();
    int getBeat(int i);
}
```