ABSTRACT CLASSES

- Any class with an abstract method is automatically abstract and must be declared as such.
- An abstract class cannot be instantiated.
- A subclass of an abstract class can be instantiated only if it implements each of the abstract methods of the super class.
- If a subclass of an abstract class does not implement all the abstract methods it is automatically abstract and must be declared as such.
- Static, private and final methods are not allowed for abstract methods since they cannot be overridden.
- Subclasses of an abstract class A can be assigned to elements of an array of A.
- You can invoke any abstract method of A for any A object despite A has no implementation of those methods.

INTERFACES

- An interface contains no method implementation.
- All methods of an interface are implicitly abstract.
- Methods should not be declared static, protected or private.
- Interfaces have no fields. The only fields they have must be declared static final (constants.)
- An interface cannot be instantiated.

INTERFACES VS. ABSTRACT CLASSES

- An abstract class may not be entirely abstract. It could implement some methods that would be useful for subclasses. An interface, however, is entirely abstract.
- A subclass can implement more than one interfaces. It can extend one abstract class only. This could cause design issues.
- If you add methods to an interface you break any class that has already implemented that interface. You can safely add nonabstract methods to an abstract class that has been already extended by other subclasses.

Reference

David Flanagan, Java in a nutshell, O'Reilly. 5th edition.