ARCTICTIGER

Chapter 7
Inheritance

Based on the course literature:

Java: A beginner's guide

Fifth Edition

Herbert Schildt

What we'll cover

- Inheritance basics
- Call superclass constructors
- Use super to access superclass members
- Understand superclass references to subclass objects
- Override methods
- Use abstract classes
- Use final
- Know the Object class

Inheritance Basics

- The subclass extends the superclass.

TwoDShape —> Triangle

```
public class Triangle extends TwoDShape{
    double area(){
        return width * height / 2;
    }
}
```

Inheritance

- A subclass can inherit from only one superclass.

- But … you can have an inheritance hierarchy.

Access and Inheritance

- private members in the superclass are not accessible in the subclass.

Constructors and Inheritance

- The superclass constructor constructs the superclass.
- The subclass constructor constructs the subclass.

- You can call the superclass constructor from the subclass.

super()

- super() is the superclasses constructor.

- When using super in the subclasses constructor super must be the first line.

```
Triangle1(double width, double height) {
    super(width, height);
}
```

super

- super also works like the key word this.
- Giving access to all the superclass members.

method overriding

```
public class A {
	void method(){
		System.out.print("A");
	}
}

public class B extends A {
	// overloading
	void method(char letter){
		System.out.print("B " + letter);
	}
}

public class C extends B {
	// overriding
	void method(char letter){
		System.out.print("C " + letter);
	}
	// overriding
	void method(){
		System.out.print("C");
	}
}
```

abstract classes

- Abstract classes are used to force a certain interface on all classes that are derived from them.
- Objects can not be created directly from abstract classes. They are incomplete
- A class extending an abstract class MUST implement all it’s abstract methods.

abstract

```
public abstract class TwoDShape {
    public abstract double area();
}
```

final

- final prevents overriding

```java
public final void method(){
    System.out.print("A");
}
```

- final prevents class inheritance

```java
public final class A {
}
```

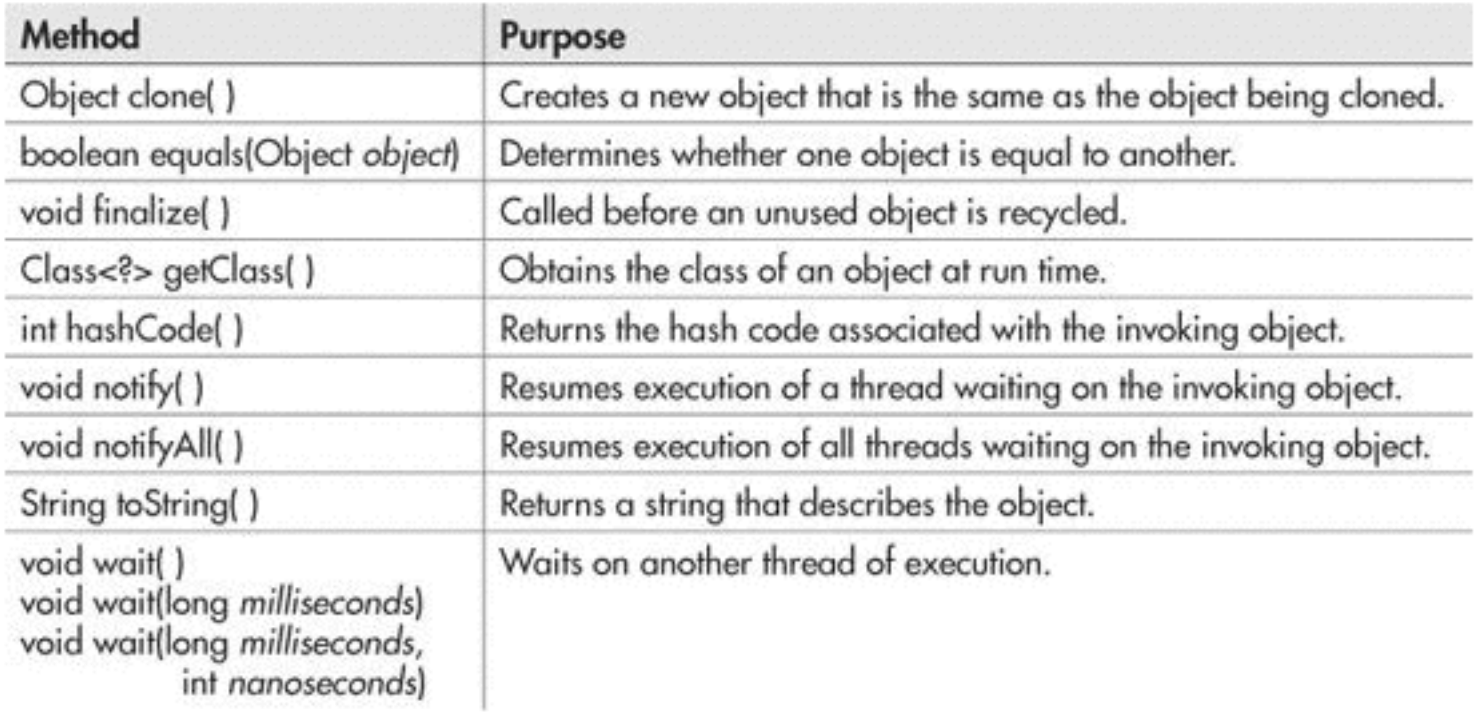
- final can be used to create constants

```java
public final double PI = 3.14;
```

The Object Class

- Object is the superclass to all objects in Java
- So every class can be assigned to a refersnce variable of type Object.
- Array is also a subclass of Object.

methods of Object



Method	Purpose
Object clone( )	Creates a new object that is the same as the object being cloned.
boolean equals(Object *object*)	Determines whether one object is equal to another.
void finalize( )	Called before an unused object is recycled.
Class<?> getClass( )	Obtains the class of an object at run time.
int hashCode( )	Returns the hash code associated with the invoking object.
void notify( )	Resumes execution of a thread waiting on the invoking object.
void notifyAll( )	Resumes execution of all threads waiting on the invoking object.
String toString( )	Returns a string that describes the object.
void wait( ) void wait(long *milliseconds*) void wait(long *milliseconds*, int *nanoseconds*)	Waits on another thread of execution.

ARCTICTIGER