Inheritance

Inheritance looks kind of interesting, but when would we really use it?

Well, it turns out, in Java, we've been using Inheritance all along.

java.lang.Object

This is because every class you create in Java, intrinsically extends a special Java class.
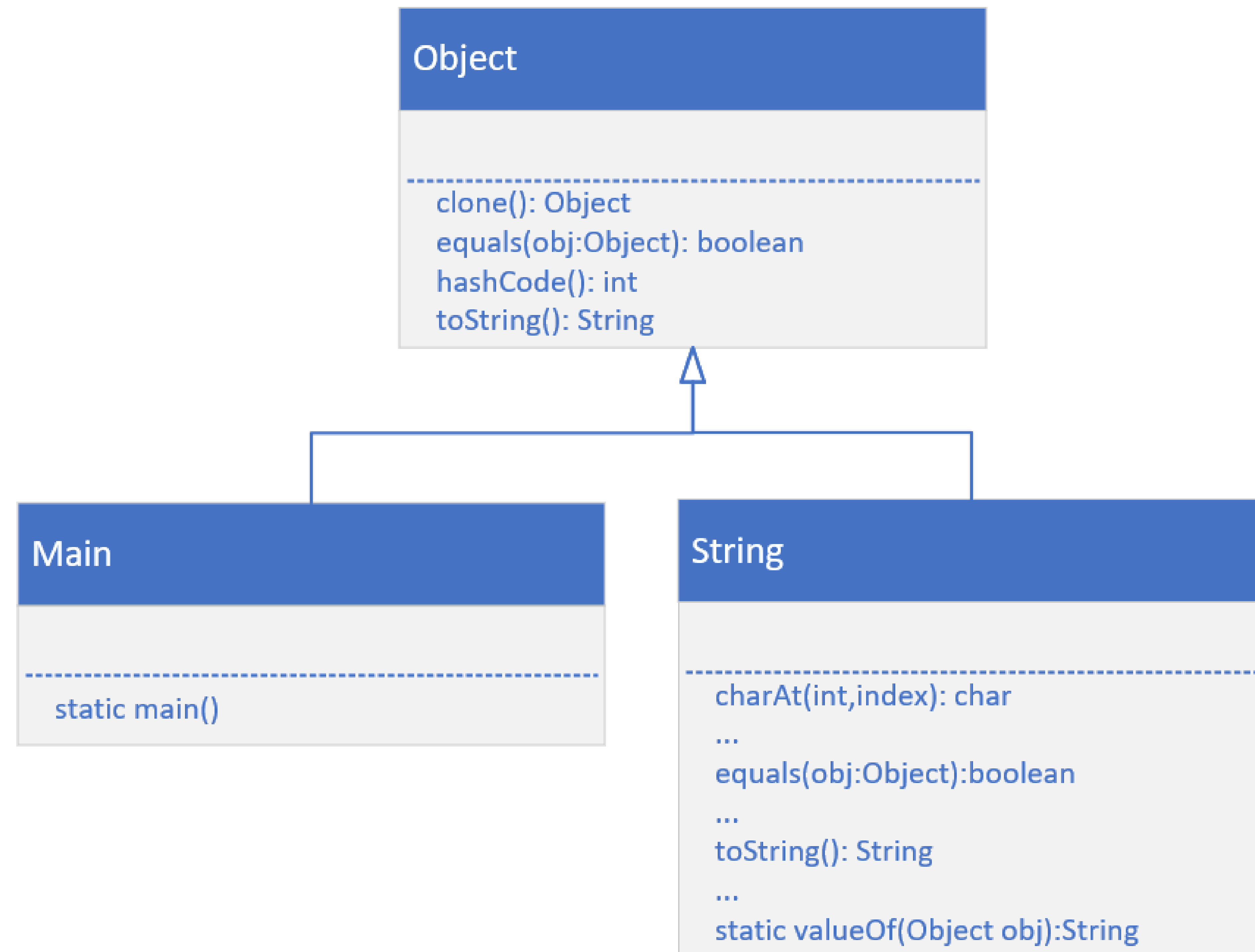
This class is named Object, and it's in the java.lang package.

Ok, that's confusing, a class called Object?

Let's see what Java has to say about this class.

We'll use the link to Java's Application Programming Interface (API) for this class, which you can find in the resources section of this video.

Every Class inherits from Object

Object
----
clone(): Object
equals(obj:Object): boolean
hashCode(): int
toString(): String

Main
----
static main()

String
----
charAt(int,index): char
...
equals(obj:Object):boolean
...
toString(): String
...
static valueOf(Object obj):String

This slide shows that our Main class inherits from, or is a subclass of Object, as also is String.

The String class has over 60 methods!

The String class overrides several methods on Object, two of which are `equals()`, and `toString()`.

Class Diagram for Student and PrimarySchoolStudent

We've already built Student, which inherits from Object implicitly.

Next, we'll build PrimarySchoolStudent, which will inherit from Student.

Object
clone(): Object
equals(obj:Object): boolean
hashCode(): int
toString(): String

Student
name: String
age: int
toString(): String

PrimarySchoolStudent
parentName:String
toString(): String