



Classes - Declaration

Short introduction



Elements of a class declaration

A class is declared in a source code file, which should have the same name as the class (including case) and the suffix `.java`

For instance, a class called `Camera` should be declared in a file called `Camera.java`

We'll briefly list the elements inside a class declaration, and we'll go deeper into each of them (most of them at least) in lectures to come.

Elements of a class declaration - package

The first element which can occur in a Java class declaration is the package statement.

The package statement is an organizational construct which lets you group related classes in “packages” by name.

Classes in a package are stored in a directory following the package name.

More on packages in a lecture to come.

Elements of a class declaration - import

After the package declaration you can have import statements. Import statements lets you use the shorter names of classes in different packages in your class source code.

Elements of a class declaration - class declaration

After the package declaration and import statements, you write the actual class declaration and class block. Example:

```
package com.company.hardware;
```

```
import com.company.storage.PictureStore;
```

```
public class Camera {
```

```
    // here's the Java code for the contents of your class
```

```
}
```

Elements of a class declaration - instance variables

In the class block, you put the contents of your class. Your class can have instance variables. Example:

```
public class Camera {  
    private int pictureCount;  
}
```

Elements of a class declaration - static variables

Your class can have static variables, shared by all class instances (objects).

Example:

```
public class Account {  
    private static double interestRate = 0.5;  
    // All account objects will have the same interest rate  
}
```

Elements of a class declaration - constructors

The constructor is the code which is run when an object of the class is created and initialized. The constructor has the same “name” as the class. Example:

```
public class Account {  
    // variables...  
  
    public Account(double initialBalance, Customer holder) {  
        this.balance = initialBalance;  
        this.holder = holder;  
    }  
}
```


Elements of a class declaration - instance methods

The instance methods define what instances of the class can “do”. Example:

```
public class Account {  
    // variables...  
  
    public void withdraw(double amount) {  
        balance -= amount; // Let's hope there's enough money!  
    }  
}
```

Elements of a class declaration - static methods

The static methods define what the class can do, totally ignoring any instances of the class. Example:

```
public class GeometryUtils {  
  
    public static double circleArea(double radius) {  
        return radius * radius * Math.PI;  
    }  
    // double in, double out - no objects are involved  
}
```

Elements of a class declaration - inner classes

Classes can actually contain other class declarations. But that's outside of the scope of the *Programming with Java* course. Example:

```
public class Customer {  
  
    class Address {  
        String street;  
        String zipCode;  
        String city;  
    }  
  
}
```

Elements of a class declaration - initializers

Classes can contain special blocks which are run only once, at defined moments. But that's outside of the scope of the *Programming with Java* course.

Example:

```
public class Inventory {  
    static {  
        readInventoryFromDatabase();  
    }  
}
```

That's pretty much it!

Now, read the wiki chapter including any links for further reading, or head on to the next chapter.

We'll discuss the elements of a class declaration in more detail in the lectures following this one.