Abstract classes

Ch 5.5
Highlights

- Abstract

```java
abstract public double distance();
```
Abstract classes

Remember classes are templates that we use to create objects

We can also use classes as templates for other classes (this is inheritance)

Sometimes when making these templates, we don't know what to fill in, but we want every class to have that method...
(See: BadCards, BadBirthday, BadHoliday.java)
Abstract classes

It is possible to create templates even when we don't know how some methods are going to work yet.

To do this we use an abstract class (shown below):

```java
public abstract class GoodCard {
```

To create the abstract class, you simply put the word abstract in your class declaration.
Abstract methods

Inside abstract classes, you can make abstract methods (an example shown below):

```java
public abstract void insideMessage();
```

You do not define abstract methods, instead you simply put a semicolon

Children of this class must override these methods (or be an abstract class itself)

(See: GoodCards, GoodBirthday, GoodHoliday.java)
Non-private abstract methods

Abstract methods must be inside an abstract class

Since children have to override these methods, they cannot be private (non-inherited)

```java
private abstract void errorMethod();
```

Error!
Using abstract classes

It is illegal to try and create an instance of an abstract class, since it is not fully defined.

You can however, use it as a type to store its children.

In fact, abstract classes are only really useful as a parent to other classes.
A larger example

A good example of where abstract classes make sense is biological classification.

All animals are broken down into:
Kingdoms, Phylum, Class, Order, Family, Genus and Species

Abstract classes make sense, because no real animal exists until the “Species” point (until then it is categorization).
A larger example

Let's look at birds!

Class: Aves
Family: Spheniscidae
Species: Fairy Penguin
Species: Loon