

CE0825a - Object Oriented Programming II 2: Inheritance and Interfaces

James A Sutherland

Abertay University

Monday, 18th January 2016

Object Inheritance

In Java, every class extends exactly one other class, `java.lang.Object` by default.

A Class is itself an object in the language: you can create a new Class – or rather, the JVM can create them for you behind the scenes.

Generics

Added in Java 5: <https://docs.oracle.com/javase/tutorial/java/generics/>

Instead of storing a list of Objects, you can be a bit more specific (and bug-free thanks to better type checking) by using generics.

For example, define a container of things: `List<?>`. If someone creates a `List<Integer>` then tries to put a `String` in it, the compiler rejects it *at compile time*.

Interfaces

- List of methods implemented
- Just a list, no definitions at all
- Zero or more methods: `java.io.Cloneable`

Abstract Classes

Like Interfaces, but ...

- Can define default method implementations
- Can't have more than one

Iterators

Handy little shortcut for iterating through lists of things:

```
for (Object o : objects)
{
    ...
}
```

Anything 'iterable' – implements `Iterable<?>` or extends `Iterator<?>`

Console I/O

Easy to generate output: `System.out.println()`

No `System.in.readLine()` though!

There is `.read()` – that only does bytes...

Wrap it in `java.io.BufferedReader`, which *does* have a `.readLine()` method.

It needs a `Reader` object though, not an `InputStream`. Enter `InputStreamReader`.

Lab Assignment 2

Write a Java class which stores a list of objects whose class extends `java.lang.Number`.

Include a method to calculate the mean (average) of the numbers in that list.

Feed numbers to it from the console. (Eclipse: `ctrl-F11` to launch, then you can type in the output window.)

Hint: `new Integer(String)`

Note exceptions, particularly `NumberFormatException` – suggestion: catch them, use that to trigger printing the mean and exiting the loop.