


Classes & Objects

A quick first look


1

Department of Computer Science 

Components

- Many manufactured things are built from collections of pre-designed components.
- A component has:
 - a *specification*.
 - an *interface* to connect it with other components.
 - well defined ways of how it should be used.


2

Department of Computer Science 

Reuse

- Components can be designed and tested once,
- and *reused* in many different contexts.
- Components can be mass-produced.

3

Department of Computer Science 

Software components

- Can we have software components?

Yes!

- A specific kind of component is described by a *class*,
- and realised by an *object*.

4

Department of Computer Science



Class

- A class describes the behaviour and implementation of a particular type of object.
- It holds design information.
- It has a name.
- It can be used to create objects.

5

Department of Computer Science



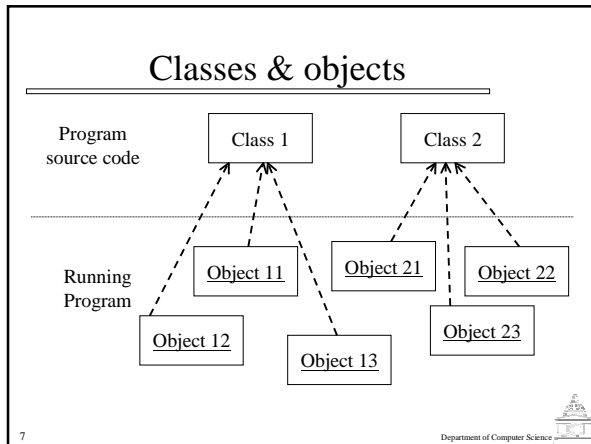
Object

- An object is a component described by a class.
- Objects are created and used by your programs.
- They provide all sorts of services.

6

Department of Computer Science





- ### Familiar objects!
-
- `System.out.print("hello") ;`
 – out is an object
 - `g.drawLine(0,0,300,300) ;`
 – g is an object
 - `String s`
 – s is an object
 - `Date d`
 – d is an object
- 8 Department of Computer Science

Remember arrays?

Yes, they're objects!

9 Department of Computer Science

Objects are everywhere!

- Java is an *object-oriented* programming language.
- It relies on using classes and objects.
- It comes with 3,000+ classes!
- You can't avoid using classes and objects.

10

Department of Computer Science 

You've already created classes!

- All your programs have:

```
public class MyClass
{
    // Your program in here
}
```
- It's a class!

11

Department of Computer Science 

So how do you use objects?

- You *call* their *methods*.
- Methods are defined by the object's class and do the work.

```
g.drawLine(0,0,300,300) ;
```

- drawLine is the method.
- It takes arguments.

12

Department of Computer Science 

Interfaces

- We say an object has an *interface*.
- You can *only* do what the interface allows you to do.
- The interface cannot be bypassed.
- The class defines what the interface is.

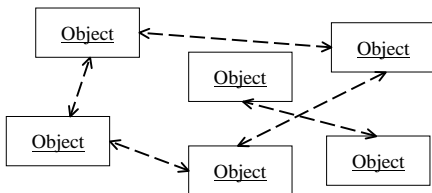
13

Department of Computer Science



Communicating objects

- Object-oriented programs consist of large numbers of communicating objects:



14

Department of Computer Science



Where does imperative programming fit in?

- It provides the underlying computational model.
- It provides the way of writing the code needed by objects.
- It provides the “glue” to join everything together.

15

Department of Computer Science



Pre-defined classes

- What classes are there?
- What methods does a class have?
 - To find out see the Java JDK documentation.
 - The “Javadoc”.

16

Department of Computer Science



The Javadoc (see Java Resources on the D0a1 web page)

Package	Description
java.applet	Provides the classes necessary to create applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating windows, interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.

17

Department of Computer Science



Creating an object

- We use *new* to create an object:

```
Vector v = new Vector() ;
Date d = new Date() ;
String s = new String() ;
Stack st = new Stack() ;
```

Class name → Vector, Date, String, Stack

Object name ← v, d, s, st
- Objects are automatically *garbage collected* when no longer needed.

18

Department of Computer Science



Importing a class

- Add
import packagename.classname
at the beginning of your program.
- Examples:
 - `import java.util.Date; //import class Date`
 - `import java.awt.* ; //import all the awt classes`
 - `import java.awt.event.* ; //import all the awt event classes`

19

Department of Computer Science



Using a class

- Import the package or the class
- Create objects related to the class
- Call class methods on the object

20

Department of Computer Science



Summary

- A first overview of classes and objects.
- They give us reusable components to build our programs from.
- At the moment we have just used them.
- By the end of the course we want to create our own!

21

Department of Computer Science