

Week 1, Lecture 2

Objects and Messages

EXPLORING
Java



Announcements

- have a floppy disk with you when in the lab
- when running Kawa make sure the **Packages** classpath is set to **C:\genesis\packages**
- in the lab the hints will be found on G drive (go to **subjects on 'Scrub'(G:)** and open the **genesis\Hints** folder

Announcements (cont)

- if you are having trouble installing Java and/or Kawa, carefully read the files **README.txt** and **OPTIONAL.txt** on our CD-ROM
- there will be a tutorial on how to install Java and Kawa, probably towards the end of next week: details will be on the web page

Week 1

Java Genesis:

–Ch1: Getting started

–Ch2: Exploring Java

Lab Assessment 1 (deadline week 5)

Java is an object-oriented programming language.

- What does this mean?
- What is an object?
- How do we make use of objects in programming?

Objects have

- **state**

- data that captures the object's current situation

- **methods (operations)**

- processes performed in response to messages received by the object

Some typical real-world objects

- chairs
- lifts
- vending machines
- cars
- people
- everything?

Some typical virtual objects

- chairs
- lifts
- vending machines
- cars
- people
- buttons, geometric objects, ...
- abstract data structures

Objects: some observations

composition

an object may be composed of other objects

association

an object may be associated with other objects

classification

an object may be similar to other objects

e.g. have similar state and methods, and so respond to messages in similar ways

Message Passing

When an object receives a message, it responds by performing an appropriate method.

As a consequence,

- the object's state may change;
- the object may send information back to the sender of the message.

Message passing: some observations

polymorphism

can send the same message to different objects

delegation

an object may send messages to many other objects

creation

an object may create a new object

rejection

an object may not understand a message

A language for passing messages

objectName . messageName (arguments)

where

objectName

identifies the object that receives the message

messageName

identifies the message being sent

arguments

is specific information sent with the message

CircleFigure class

some messages:

- **create()**
- **moveRight(x)**
- **moveTo(x, y)**
- **getXCentre()**
- **setColour(c)**
- **drawHollow()**