Software and Programming II

Introduction and overview

Keith Leonard Mannock

Department of Computer Science and Information Systems
Birkbeck, University of London

keith@dcs.bbk.ac.uk

Tuesday, 30th September, 2014

Welcome to the Software and Programming II module

`http://moodle.bbk.ac.uk`

What is this course about?

- This is a programming course at the intermediate level
  - The language we are using is Java
  - We will use the Java Virtual Machine (JVM)
- The primary audience is BSc ISM/ISC students
  - It follows on from Software and Programming I
  - Please note the pre-requisites
- It is also a service course for other students who need to learn an object oriented programming language

Who are the lecturing staff?

Keith Mannock

- keith@dcs.bbk.ac.uk
- www.dcs.bbk.ac.uk/~keith
- x6713 MAL157

Oded Lachish

- oded@dcs.bbk.ac.uk
- www.dcs.bbk.ac.uk/~oded
- x6862 MAL166

What are you getting yourself into?

- Programming is intellectually challenging
  - It can be tremendous fun
    if you like that sort of thing!
- Lifelong learning is essential (*learn-2-learn*)
  - The technology is constantly changing
  - We cannot teach you all you need to know
  - We can point you in the right direction and give you a good, hard push - but the rest is up to you!

Programming can be fun

- Programming is puzzle-solving
  - Very little is mechanical, routine work
  - You always have to be thinking
- If you like solving puzzles, theres a good chance you will like programming
  - Some puzzles are hard
  - You need a tolerance for frustration
  - Solving hard puzzles can be very satisfying

Elegance in programming

Consider the following problem:

- You want to find the largest of three numbers
- How do you do this?

- You want to list three numbers in ascending order
- How do you do this?

Changes in computer science

Computer science is only about 60 years old

- Java was first introduced in 1994!
- We will be covering features of the language that did not exist this time last year

Change is rapid and accelerating

- Dominant language of the 1990s: C++
- Dominant language of early 2000s: Java
- Dominant company: IBM to Microsoft to Google to Apple to
- First GUI: Macintosh, 1984
- First web browser: Mosaic, 1992
- Web pages: HTML to DHTML to XML
- ...

Topics we will try to cover...

- Review of Java programming constructs
- Unit testing and mocking
- Exception handling and errors
- Arrays and Lists
- Objects and Classes
- Inheritance and Polymorphism
- Closures (probably no time)
- Collections
- GUI programming
- Web programming (probably no time)

but programming isn't just about language features and topics...

Small projects

You can build a kennel in a few hours

- You dont need a blueprint
- The materials dont cost much
- A little knowledge of tools is enough
- Imperfections are no big deal

Medium-sized projects

You can build a house in a year or so

- You really do need blueprints
- Excess materials mean wasted money

- House building requires more skills: plumbing, bricklaying, electrical work, carpentry, etc.
- Imperfections matter: you dont want a leaky roof!
- Its easier if you arent doing it all by yourself

Large projects

You cannot build a skyscraper by yourself

- Its just too much work for one person
- You dont have the money
- You dont have all the skills
- Imperfections could be costly or even fatal

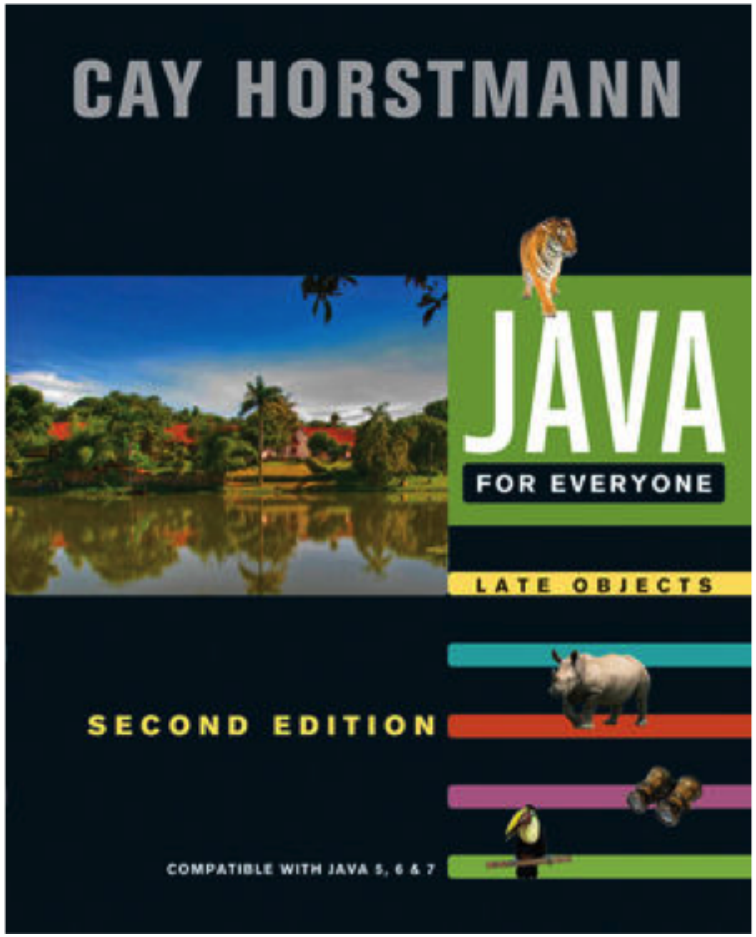
Skyscrapers can only be built by a team

- Communication is essential
- A “paper trail” is essential

What does that mean for you in this course?

- What can we ask you to build in your classes?
- What will be expected of you in industry?
- We teach skyscraper-level skills, but
  - we ask you to apply those skills to kennels
  - its silly, but what alternative do we have?
- Its up to you: When you leave here,
  - will you be able to build skyscrapers?
  - or will you just be very good at building kennels?
  - I know what Id prefer

Textbook - if you want one



Software. . .

- The Java Development Kit (JDK) version 1.8
- The documentation for the JDK
- JavaFX Scene Builder
- The Integrated Development Environment (IDE) of your choice (although you can use the command line if you really want to and/or a text editor)

Labs I

For the labs, well . . . you are sitting in them! (MAL404)
Most of the time we'll switch into "lab mode" around 8pm although we will move the session around depending on how things are going each week.

- We are using the DCSIS labs
- This means we have a good range of tools available

Labs II

Everyone is expected to attend lab, but...

- If you are already a programmer, and you understand the material on the worksheet, then you probably do not need to attend
- No new material will be presented in the lab sessions; some demos and tooling, but no new language features
- The "lab sheets" can be done at home (or completed there)

Assessment

- By three hour written examination (75%)
- By practical coursework (25%)
  The practical coursework consists of programming assignments
- Submission will be via Moodle
- The late policy is as stated in the degree handbook; the cutoff date is 14 days after the due date

How to get a good grade?

How to get a good grade? I

- Start your assignment(s) early!
  - This is the first and most important way to improve your grade
  - Programming takes a lot of time
  - Its not easy to predict how long a program will take
- Do as many exercises as you can (oh, and don't look at the answers until you've had a good go at the problem)

How to get a good grade? II

- Test your programs thoroughly
  - One or two simple tests are not enough
  - We often provide simple but incomplete tests, just to get you started
  - We will do thorough testing, even if you dont!
- Read the assignment carefully
  - Do what is assigned, not *something like* what is assigned
- Learn to use your tools (e.g., eclipse, JUnit, etc.)
- Use comments and good style right from the beginning, not as a last-minute addition
- Review and understand the lectures

Who to ask when it all goes horribly wrong!

If you have questions about the course:

- Asking your colleagues is acceptable
- Looking of forums (e.g., StackOverflow)
- Asking on the forums in moodle
- Ask or email the TA, Keith, or Oded

Keith and email...

Feel free to send me email at:

`keith@dcs.bbk.ac.uk`

I get lots of spam, including several virus-carrying messages a week, so I run several filters.

To avoid my spam and virus filters:

- If your email concerns this course, please put SP2 2014 somewhere in the subject line

Questions