UCL

COMP1007

Principles of Programming

UCL

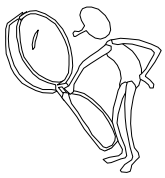
Part I - Imperative Programming with Java

Dr. Graham Roberts
email: G.Roberts@cs.ucl.ac.uk
url: http://www.cs.ucl.ac.uk/staff/G.Roberts
office: room 6.17

UCL

Agenda

- Introduction.
- How to study.
- Course details.
- Problem class and lab arrangements.
- Assessment.
- Your commitment.
- Some thoughts...



UCL

Programming in the 1st Year

- Term 1, 1007 – Principles of Programming.
  - Part I: Imperative programming with Java.
  - Part II: Declarative programming with Prolog (to be taught by Mohamed Ahmed).
- Term 2, 1008 – Object-oriented programming.
  - With Java.

UCL

Learning to Program

- The best way to learn how to program is to write lots of programs!
- Problem classes.
- Lab classes and exercises.
- Lectures support and expand but also explore a wider range of subjects.

UCL

What do you need to know to get started?

- No previous programming experience is assumed – we start from the beginning.
- You do need to be familiar with using the workstations.
- It also helps a lot if you practice your typing skills.
- What if you do have programming experience?
  - Talk to me.

Studying at University

- This applies to all courses you take.

Basics…

- This is not school.
- You are in charge of your life.
- Plan and use your time effectively.
- Learn how to learn.
- Pace yourself.
- Don't get over-tired.
- Enjoy your time here.

How much work?

- Equivalent to a full time job
  - 35 to 40 hours per week minimum,
  - some overtime.
- Time to relax, earn money, etc.
  - But your degree should always come first.
  - Be prepared to drop over things in favour of academic work.
  - Keep a balance.

Spare time between lectures?

- Use gaps between lectures/classes well.
  - Work in labs.
  - Study in the library.
  - Student common room.
- Don't spend all the time in the Union, Clubs and Pubs…

Why have lectures?

- To structure a course and set the pace.
- To tell you what you need to learn.
- To hear opinions.
- To see examples.
- A chance to ask questions.

Attending Lectures

- You need to attend lectures – don't assume the lecture material will be available anywhere else.
- Some lectures are used for tutorial/problem solving sessions.
- Check timetables regularly.

What you do in lectures?

- Stay Awake and Listen!
- Think!
- Do ASK QUESTIONS.
- Take notes.
- Turn off mobile phones.

Behaviour

- Please DON'T talk, whisper, or fidget.
- I mean that…
  - You'll be asked to leave if you can't behave reasonably.
  - Unsatisfactory behaviour can lead to you being removed from the course.
- Feel free to tell other students making a noise to shut up.

Poor hearing/eyesight?

- Let the lecturer know.
  - Ask for the microphone to be used.
- Sit at the front.
- Help available in the college.

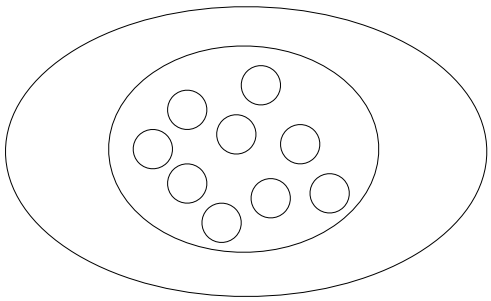
Note Taking

- You will get copies of some lecture slides
- BUT, do make additional notes to help you remember what was said.
- Not everything I say will be on a slide.

After lectures?

- 1 lecture typically leads to 2-3 hours further study.
- Review notes.
- Study topics introduced.
- Read.
- Do exercises and coursework.

The Subject



Study Strategies

- Must spent time reading, practicing, programming outside lectures.
  - Full-time occupation.
  - Immerse yourself in the subject.
- Study groups.
- Use problem and lab classes effectively.
- Look for depth, don't simply hunt marks.

3 stages of learning

- Rule follower (1st year)
- Problem solver (2nd year)
- Expert (3rd year?)

- Learn to select and evaluate possible solutions.
- Learn how to solve problems without relying on just following rules.
- Learn how to assess yourself.
- Learn to recognise good quality work.

"Knowledge is only part of understanding. Genuine understanding comes from hands-on experience."
Dr. Seymour Papert, MIT
(LEGO Mindstorms)

Student Representatives

- I need nominations.
- Email me.

Back to 1007…

Lectures

- Thursday 2-3 noon Lankester LT (Now!)
- Thursday 4-5 Lankester LT
- Friday 11-12 Archeology LT
- Friday 2-3 J Z Young LT (Anatomy)
- Check the timetable.

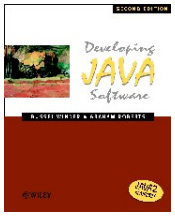
Email Registration

- Make sure you register on the 1007 mail list.
- Send an email to 1007-request.
- Type join on the subject line.

- Only register from a CS dept. machine with a CS email address.

Recommended Course Text Book

Developing Java Software, 2nd Edition
by Russel Winder and Graham Roberts
pub. by John Wiley & Sons (2000)
ISBN: 0-471-60696-0



- The book contents are the lecture notes for this course.
- It also contains the Java language reference you need.

Where is the bookshop?

- Go to Waterstone's Book Shop.
- The computing section is in the basement.
- Or go to amazon.co.uk.
- Or visit the Science Library.

Problem Classes

- You are allocated to a weekly problem class.
- Run by Research Fellows.
- Problems handed out week before to think about.
- Solve problems, discuss, get feedback.

Why?

- To learn how to solve abstract problems.
- The essence of Computer Science.

Example?

- A regular hexagon has six sides. Given the coordinates of the centre of a hexagon and the length of a side, how do you calculate the coordinates of each corner of a hexagon?
- Easy!?

Lab Groups

- You are allocated to a lab group of around 10-15 people + a demonstrator.
- Each group has a weekly lab class.
  - You attend one of the timetabled labs per week.

What happens in a lab class?

- All lab classes are held in 1.05 lab.
- You work on your programming exercises.
- Demonstrators will be present to:
  - give you help and advice while you program.
  - give you feedback.
  - do talk to them.

Do you only write programs in the lab class?

- No – you can and should use the lab any time it is not being used for timetabled classes (or use your own computer).
- Lab classes are where you can get direct, “in front of the screen” help.

Why have labs at all?

- Because:
- It is your responsibility to learn Java and develop your programming skills but you need proper support.
- Lectures can only give you so much knowledge – the rest has to come from practice and experience.
- You have to start serving your apprenticeship!

Using your own Computer

- You can use your own PC/Mac for doing programming exercises.
- It is your responsibility to maintain your computer and back-up data.
- “My computer is broken” is not an excuse!!
- (The facilities we provide are entirely adequate – owning a computer is useful but not essential.)

Where do you get the Java software?

- We use the version of Java called:
- “The Java 2 Platform, Standard Edition”.
- We have installed J2SE version 5 (Java 5).
- You can get a CD for Windows, GNU/Linux, OS X from from the departmental office (cost £1).
  - Or download yourself (java.sun.com).
- Macs already have Java installed but you will need to update it to Java 5.

Prolog?

- Mohamed Ahmed will cover books, software, etc. later.
  - Second half of term (last 5 weeks).

Short Pause...

- Any Questions?

Assessment

- The course is assessed by both exam and coursework.
- The final exam counts for 90% of the overall mark.
- The coursework counts for 10% of the overall mark.
- The minimum overall pass mark is 40%.

Passing the course

- The exam and coursework components are assessed independently (each out of 100%).
- You must separately pass both.
- The minimum pass mark for each component is 40%.

The Final Exam

- Held during term 3 – the exam term (April/May 2006).
- Lasts 2.5 hours.
- Has questions on programming and problem solving.

Example exams papers?

- 1007 started last year (2004/5).
- Replacement for 1B1A in 2003/4 (and part of 1B11 in earlier years).
  - Similar content, though.
- Look at 1B1A and 1B11 past exam papers for example Java questions.

1007 Mid-Session Exam

- Held in early January 2005.
- Allows both you and us to assess your progress in 1007.
- Doesn't count towards your final mark.
- Based on the results we will identify those needing extra help.
- We expect everyone to pass without problem.

Programming Coursework

- There are 3 kinds:
  - Exercises
  - Coursework Tests
  - Mini-projects

Exercises

- Sets of exercises will be handed during the course.
- Notes, examples and questions.
- Core questions must be answered if you are keeping up.
- Additional questions can be done to push yourself forward.

Exercise Evaluation

- Not formally assessed.
- But do give essential experience.
- Ask your demonstrator to give feedback during lab sessions.

Coursework Tests

- Held every 5 weeks.
- Must attend! Dates will be pre-announced.
- Mixture of multiple choice and short answer questions.
- If you are absent see the Departmental Tutor ASAP (i.e., me).

Mini-projects

- Graded A-F.
- You design and write a larger program.
- First one in Java due in after Reading Week.

Only 10%?

- Total coursework value (with respect to the final course mark) is 10%.
- Tests monitor your progress.
- The mini-projects sees how you tackle a larger problem.
- A small incentive but not about chasing marks.

Plagiarism

- Copying someone else's work.
- Cheating.
- Don't do it.
- If you get caught the consequences are serious.

Discussing programming

- It is alright to discuss programming with others.
- In fact, it is important – programmers need good communication skills.
- But don't simply copy answers.
  - You will be the loser as you won't learn how to solve programming problems.

Short Pause

- Any questions?

Is this a hard course?

- Yes!

Well, I would say that…

BUT I'm not joking.

Your Commitment

- 1007 is rated at 150 hours work.
  - 30+ hrs lectures
  - 60+ hours exercises/coursework
  - 60+ hours reading, practice and revision

How do you pass?

- Keep working methodically.
- Keep practising your programming.
- Read the book(s).
- Don't slack off.
- All the stories you've heard about university life being easy are false – you have to work hard. This is UCL.

You're Driving…

- You need to have the self-discipline to work and study hard.
- This is not a school – you have to drive your own progress.

What if you don't pass the course?

- You cannot enter the 2nd year without passing this course.
- You can try again next year but would have to take a year out while waiting.
- There is no guarantee of summer resits.

Don't Panic!™

- We want you to pass.
- There are plenty of people to ask for help if the going gets tough – your demonstrator, me, your tutor, and others.
- You CAN do it.

What makes a good programmer?

- Being logical.
- Perseverance.
- Boldness.
- Attention to detail.
- Lots of practice.
- Reading the literature.
- Experimentation.

Relationship to other courses

- 1B10, 1B12, 1B13 – look for the links and connections.
- Next year 2007 will continue on from 1007/8. 2009/10/11 also include programming work in Java or related material.

Done!

- That's the course introduction.
- Questions?

Something to think about

- What is Science?
- What is Engineering?
- Will you actually be doing either of these activities?