ELEC0021 - PROGRAMMING II
OBJECT-ORIENTED PROGRAMMING

Overview and Introduction to Java

Prof. George Pavlou
Communication and Information Systems Group
Dept. of Electronic and Electrical Engineering
http://www.ee.ucl.ac.uk/~gpavlou/
g.pavlou@ucl.ac.uk

Syllabus

Programming I taught basic programming via C

Programming II focuses on more advanced object-oriented programming using Java

- Introduction to Java, classes & objects
- Random numbers; Exceptions
- Class encapsulation and inheritance
- Data structures – List, Stack, Queue
- Algorithms and complexity
- Recursion
- Data structures 2 – Binary Trees
- Threads and concurrency

Assessment

80% Examination in Term Three

20% Programming Assignments
- No assignment mark, no module mark!

Reading List

There are no compulsory texts for this course. The following book gives a good introduction to Java and object-orientation

Java language

- H. M. Deitel, P. J. Deitel, Java: How to Program, Pearson / Prentice Hall
  - Fifth Edition 2004, compliant with J2SE 1.4.1
  - Sixth Edition 2005, compliant with J2SE 5.0
  - Later editions
- There also exist plenty of resources and books online

C and Java

- In the 1st year you were taught programming in C with a very brief introduction to Java
  – Target was to familiarise yourselves with programming
- This year focuses on object-oriented (O-O) programming through Java
  – Target is to advance and learn good programming principles
- C is a powerful language that allows full manipulation of the underlying memory
  – The same also applies to C++ and Objective-C which are object-oriented C “extensions”
- Java is an O-O language but takes care of memory management, and as such, it is <u>much easier to use</u>
  – Increased productivity, easier to implement complex programs
  – But not suitable for low-level, high-performance programming

Object-Orientation

- In non O-O languages, code is structured through functions, procedures or subroutines
  - functions in C, procedures in Pascal, subroutines in Fortran
- A re-usable piece of code that takes parameters, does something with them and then returns a result
  - E.g. long square (int n) { return n*n; }
- Data items needed by many functions are typically global i.e. accessible by all program functions
  - Potentially prone to errors, make functions not easily reusable
- In O-O languages, code and data are tightly coupled into object classes, with object instances creating data copies
  - The object data represent "state": instance variables
  - The object code operates on the instance variables: methods (a method is a function that has access to the object data)

Platform Independence

- Most languages are compiled to the assembly language of the underlying microprocessor
  - Compiled code is only executable for the device it was compiled for (i.e., processor + operating system)
- Java code is compiled to an intermediate representation known as bytecode which is “platform neutral”
  - Bytecode is the language of the Java Virtual Machine (JVM)
- The JVM itself is compiled for a specific device and can then run any Java code compiled anywhere else
  - This gives Java platform independence i.e. portability
- This approach results in relative inefficiency and, together with automatic memory management, they make Java non suitable for real-time embedded programs
  - But is fine for the vast majority of software applications
  - Benefits: easy to use, increased productivity, ubiquity

Java and Other Languages

- C and Java form a powerful couple of base languages allowing one to deal with any programming in the future
  - C teaches/supports low-level memory management while Java teaches/supports structured programming using O-O principles
- C++ is an O-O language <u>compatible with C</u> (similar)
  - Design influenced by Simula, focus on performance like C
  - Not <u>pure</u> O-O like Java and Objective-C, good for embedded s/w
- Objective-C is an O-O <u>extension to C</u> (superset)
  - Design influenced by Smalltalk (like Java), used by Apple
  - Swift emerged in 2014 for better performance and safety
- C# is Microsoft’s “Java” in its .NET initiative
- Python is an <u>interpreted</u> language that supports quicker “prototype” programming than <u>compiled</u> C, C++ and Java