School of Mathematics

Course 262 (2E3) — Computer Science 2 (SF Mathematics)

Lecturer: Dr. Steven Collins

Requirements/prerequisites: 161

Duration: 24 weeks

Number of lectures per week: 2 lectures, 1 tutorial and 1 laboratory

Assessment: 40% for continuous assessment of several kinds.

End-of-year Examination: 3-hour exam in May/June.

Description:

The course covers advanced topics in C++ programming, approaches to program design, basic static and dynamic data structures, searching, sorting, and introduces some standard algorithms. The module provides a thorough grounding in the principles of object-oriented design principles, introducing concepts including abstraction, inheritance, polymorphism and encapsulation. There are 2 lectures, 1 tutorial and 1 laboratory per week. During the year there are 2 major projects giving an opportunity to put into practice the design skills studied. In addition, students are given the opportunity to make presentations to their peers on a variety of topics relating to the impact of computers and software engineering on society. To complement this outside speakers are invited from research groups and from industry to provide additional context and insight to the students. The course has an emphasis on continuous assessment which counts for 40% of the final mark, with the remaining 60% determined by the end of year exam. The coursework mark is composed of marks awarded for tutorials (7 or 8 tutorials during the year are collected and examined) and for the projects (typically a first smaller project due in the first term followed by a larger project due in the final term).

There is a course web page at http://isg.cs.tcd.ie/scollins/moodle/ In some detail, the contents are as follows:

- Revision of basic concepts of computer programming. Selection (if statement), case statement, while/for/do while loops, conditional expressions, terminal I/O, simple data types, enumeration, arithmetic relational & logic operators, type casting, expressions (C++).
- Fundamental Concepts of Object Orientation: Concept of class, object instance, private and public members, methods, functions, parameters, local and global variables, void and value functions, simple example programs using classes.
- Object Oriented Programming using C++. Encapsulation, class specification and implementation, scope operator, building up class libraries, software reuse, functions revisited, scope rules, parameter passing by value and reference, recursion, overloading of functions, function templates, software re-use.
- Classic Data Structures, Data Representation and Algorithms. Arrays, (1 & 2 dimensional), array processing, search & sorting algorithms, stacks, calculation of performance

2006-07

(complexity theory).

- *Object Oriented Software Techniques:* object oriented design methodology, static variables, inheritance, class templates, class libraries.
- Complex Data Structures: Dynamic data structures, pointers, queues, stacks (implemented with pointers), trees.

Textbooks:

- Savitch, W., Problem Solving With C++: The Object Of Programming, Sixth Edition, published by Addison Wesley, ISBN 0-321-44263-6. TCD Library Reference: 500.16424 N93*4;1-9 (5th Edition)
- Malik, D.S., C++ Programming: Program Design Including Data Structures, Third Edition, published by Thompson, ISBN 978-1-4188-3640-5. TCD Library Reference S-LEN 500.16424 P69;1-11

September 25, 2006