

122COM: Module information

David Croft

Coventry University

david.croft@coventry.ac.uk

2017

Overview

1 Module structure

2 CodeAnywhere

3 Recap

10 credit module. Full coursework evaluation

- MCQ Week 5 (20%)
- MCQ Week 10 (20%)
- ALL project (60%)
 - Alternative project for retake students.
 - Alternative project for Jan starters.

- 1 Algorithms
- 2 GUIs
- 3 C++ intro
- 4 Searching algorithms
- 5 Profiling and complexity
- 6 Database integration
- 7 Memory
- 8 Data structures
- 9 Sorting
- 10 Testing

Order of topics may change.

New topic introduced each week with lecture. Accompanied with practical coding exercises to be completed during the lab.

- Pre-lab work each week will be set (same as 121COM).
 - Not optional (same as 121COM).
- Material will be marked using traffic light system (same as 121COM).
 - Understanding green material required to pass module.
 - Understanding yellow material should produce good mark.
 - Red material is advanced, for students with previous coding experience or students looking to stretch themselves. It is not testable.

Assuming you have working knowledge of Python3.

C++11 introduced during the module

- Majority of students expected to work in C++.
- Tested on C++ syntax during phase tests.

BIT/ITB and Multimedia Computing students

- Expected to complete some C++ labs
- Will not be asked questions regarding C++ syntax during phase tests.
- Can be asked generalised questions re. differences C++ & Python.



- Write GUI and/or database integration code.



- Write GUI and/or database integration code.
- Write and understand algorithms.
 - Searching algorithms.
 - Sorting algorithms.
 - Algorithmic complexity.



- Write GUI and/or database integration code.
- Write and understand algorithms.
 - Searching algorithms.
 - Sorting algorithms.
 - Algorithmic complexity.
- Write some C++ code.
 - BIT & MC will not be tested on C++ syntax.

- Write GUI and/or database integration code.
- Write and understand algorithms.
 - Searching algorithms.
 - Sorting algorithms.
 - Algorithmic complexity.
- Write some C++ code.
 - BIT & MC will not be tested on C++ syntax.
- Understand memory.
 - Pointers.
 - Stack and Heap.

- Write GUI and/or database integration code.
- Write and understand algorithms.
 - Searching algorithms.
 - Sorting algorithms.
 - Algorithmic complexity.
- Write some C++ code.
 - BIT & MC will not be tested on C++ syntax.
- Understand memory.
 - Pointers.
 - Stack and Heap.
- Understand data structures.
 - Stack.
 - Queue.
 - Array.
 - Vector.
- Understand testing.
 - Automated testing.

The module runs in one semester over 11 weeks.

- 2 hours contact time a week as single lecture/lab block.
 - Students expected to spend additional 4 hours self study.
- Additional support available as part of programming support centre
 - <https://gitlab.com/coventry-university/programming-support-lab/wikis/home>

There is no required reading for this module.
However, the following resources can be recommended.

■ Python

- Learning Python, 5th Edition. Mark Lutz, O'Reilly.
- Automate the Boring Stuff with Python, Al Sweigart. Free PDF version online.
- The 121COM material and reading list.

■ C++

- C++ Programming In Easy Steps, 4th Edition. Mike McGrath.
- Penguin programmer - An excellent beginners C++ guide.
<http://www.penguinprogrammer.co.uk/c-beginners-tutorial/>.
- Learncpp - A more advanced C++ guide that goes into greater depth.
<http://www.learncpp.com/>



Using Codeanywhere (<https://codeanywhere.com/>) for this module.

- Starting next week.
- Sign up for free account.
 - Before next session.
- Anyone unable to sign up for Codeanywhere must email David before Saturday.
- Recommend you create a single C++ Ubuntu 14.04 container and reuse it throughout module.

Recap

- Sign up for Codeanywhere
- Learn how to use Codeanywhere

The End