Advanced Programming Techniques Introduction

COSC1076
Semester 1 2020
Week 01

RMIT UNIVERSITY

Course Introduction

RMIT UNIVERSITY

Course Introduction

- This course is about ***programming concepts and skills***, including:
  - Representation of data types
  - Pointers, references, and concrete types
  - Primitive types vs objects and classes
  - Dynamic Memory management
  - Advanced operator overloading, and polymorphism
  - File processing
- These concepts and skills are largely language independent*
- It is also good for software engineers to gain experience in using different programming languages
  - You will also learn to use the C++11/14 programming language as well as the basics of the Unix operating system.

Course Introduction

- This course is also about ***software engineering***, in particular:
  - Correct and error free software
  - Having a complete understanding of any block of code
  - Static code analysis
  - Good coding style
  - Ethics of software engineering and software development

Be a Software Engineer
Do it right the first time

RMIT UNIVERSITY

Mistakes are *errors* not bugs

RMIT UNIVERSITY

Teaching Team

- Course Co-ordinator / Lecturer
  - Timothy Wiley
  - Consultation hours: Mondays 3.30pm - 5.30pm
- Tutors
  - Dale Stanbrough
  - Jake Weiner
  - Robert Tirtasentana

Assumed Knowledge

- This course assumes you have done 1 to 2 semesters of programming in Java or another programming language
- We assume you are familiar with:
  - Foundations of programming: Basic control structures, Common data types, 1D and 2D arrays, Functions
  - Foundations of Object Orientated programming: Classes, Inheritance, Polymorphism, Public interfaces, Concepts of Abstraction in software design, Modular program design, Encapsulation
  - Good coding style
  - Good software engineering practices
- If you don’t know these concepts already, you are in the wrong course

Expectations

- This course is ***intense and challenging***
  - It is more difficult than previous editions, by choice of CSSE/CSIT
  - This course is ***core*** for CS, SE, and Games students
  - IT students:
    - This is an Advanced Elective
    - It will be one of your hardest courses
    - However, if you find it too difficult please make sure you change courses ***by the end of Week 2***

Expectations

- Us -> You
  - Expect you to work hard and be highly motivated
  - Designed to be Face to Face, please attend classes
  - Interact while in class! Be curious, ask questions.
  - Follow standards of Academic Integrity and Ethical Behaviour
- You -> Us
  - Strive to be prompt in replying to your requests
  - Provide support and help for you to achieve the best learning outcomes as you can
  - Be fair and allow you opportunities to ask us about assessment marks, but please do not haggle about them

What your Grade means

PA	▶ You are learning the skills & knowledge of this course. ▶ Keep working hard, so that you don't fall behind. ▶ We encourage you do put in a bit more effort.
CR	▶ You are right on track ▶ Keep up the good work
DI	▶ You are working at an above average level ▶ You are putting in some extra work and extending your skills
HD	▶ You have gone well above and beyond the requirements for this course. ▶ Make sure your other courses are going just as well!

Syllabus

Getting Started with C++	Dynamic Memory Management II
Strings, Classes, Pointers & References	Dynamic Data Structures & Recursion
Dynamic Memory Management I	Advanced Polymorphism and Typing
Version Control & Program I/O	Advanced Topics
Abstraction & Containers	

Course Resources

- Lectures are recorded on Echo360
  - Not necessarily a substitute but good for revision
- Reading List
  - On Library - links to:
    - E-book references
    - LinkedIn Learning Videos on related concepts
  - Updated week-to-week
- Text book(s)
  - Stroustrup, B., ***Programming principles and Practice using C++***, 2nd Edition, Addison-Wesley, 2014, ISBN 978-0-321-99278-9

Course Resources

- Software
  - G++ - Compiling C++ - cross platform tools available
  - VSCode - Code editors (Not necessarily IDEs!)
  - Git - For code repository (BitBucket)
  - SourceTree - Observe Git repository
- Forum
  - Ask questions here preferably
- Lynda Modules
  - For extra revision

Class Structure

- Lectures
  - More like workshops
  - Live worked examples

- Tutorials/Labs

- Consultation
  - Online: Mondays 2.30pm - 3.30pm
  - In-person: Mondays 3.30pm - 5.30pm

- Self revision questions

Assessments

- Assignment 1 (15%)
  - Individual
- Assignment 2 (30%)
  - Group
- Lab Exercises Weeks 2-6 (5%)
- Exam (50%)

Academic Integrity

- Plagiarism & Academic Integrity are very serious
  - However, ***copying other people doesn't help you learn!***
- You are encouraged to form groups to solve problems
  - However, when writing or programming, write in your own words or code, and provide your own solutions.
- This is not about penalising or yielding the stick. You've made a conscious act to enroll in this course, and we want you to:
  - Learn by trying things, making mistakes
  - Fair for everyone
  - Quality control of your degree
- More info on Canvas

RMIT
UNIVERSITY