Course Outline Public

Edit Resource (/COMP1531/19T3/resources/31863/edit)

Contents

- Course Details
- Course Summary
- Assumed Knowledge
- Student Learning Outcomes
- Teaching Strategies
- Teaching Rationale
- Student Conduct
- Assessment
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Code	COMP1531
Course Title	Software Engineering Fundamentals
Convenor	Hayden Smith (https://webcms3.cse.unsw.edu.au/users/z3418003) , Rob Everest (https://webcms3.cse.unsw.edu.au/users/z3254687)
Admin	Hayden Smith (https://webcms3.cse.unsw.edu.au/users/z3418003) , Rob Everest (https://webcms3.cse.unsw.edu.au/users/z3254687)
Classes	**Lectures** : • Monday 4pm-6pm, CLB7 • Thursday 4pm-6pm, CLB7 Timetable for all classes (/COMP1531/19T3/timetable)
Consultations	• Monday 6pm-7pm, in the foyer outside CLB7 • Thursday 6pm-7pm, in the foyer outside CLB7
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs1531/19T3/ (http://cse.unsw.edu.au/~cs1531/19T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1531.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1531.html)

Course Summary

This course is teaches students about software engineering principles via exposure to the important practice of building correct products in effectively functioning teams.

The students are exposed to agile software practices, team collaboration and effective communication through implementing a group project based on agile software methodologies that requires them to analyse, design, build and deploy a web-based application. This course is typically taken in the semester after completing COMP 1511, but could be delayed and taken later. It provides essential background for the teamwork and project management required in many later courses.
The goal of this course is to expose the students to:

- Demonstrating effective use of the python programming language
- Basic elements of software engineering - derived from the life-cycle of a software system, including requirements elicitation, analysis and specification; design; construction; verification and validation; deployment;
- Software engineering methodologies, processes, tools and techniques
- Agile software, collaboration and communication practices
- Web-based system architecture and development practices on web platforms

Assumed Knowledge

We assume all students have completed COMP1511 (or equivalent).

Students should be familiar with the basic concepts of programming, including loops, functions, libraries of code, compiling, and writing code to follow specifications.

Student Learning Outcomes

After completing this course, students will be able to:

1. Identify the complexities of software design and development
2. Analyse a problem domain and elicit user requirement, user stories, and use cases
3. Understand modern tools available to engage in deployment of web-based applications
4. Demonstrate effective use of applying the python programming language to solve problems
5. Demonstrate proficiency in use of system schematics, data modelling, and conceptual modelling, to analyse complex software systems
6. Demonstrate effective usage of testing fundamentals (e.g., unit tests, integration tests, test plan/cases, test automation)
7. Describe the phases of software development and life-cycle of software - and illustrate them from experience
8. Reflect on the choice of software engineering methodology (e.g, waterfall, agile) used in a project
9. Understand agile software development practices and conducting analysis, design, implementation and testing in the context of an agile framework
10. Describe common behavior that contribute to the effective functioning of a team and identify necessary roles in a software development team based on agile software practices
11. Understand and apply Git as a source code management tool and as a medium for collaborating in the development of software applications

Teaching Strategies

This course uses the standard set of practice-focused teaching strategies employed by most CSE foundational courses:

- Lectures
- Tutorials and Lab sessions
- Major Project

This course aims to provide the students with a strong foundation in the fundamental principles and practices of software engineering that will prepare them for the advanced software engineering workshops. As such, a broad range of key software engineering topics will be taught and reinforced through a group project, that will enable students to apply the theoretical concepts acquired to solve a practical software engineering problem. An agile software delivery style has been chosen for the implementation of the group project, to make students familiar with modern agile development methodologies.

Each tute-lab session will comprise a 1 hour tutorial followed by a 2 hour practical lab session. These sessions **commence** from **week 1** .

Lectures

Lectures will be used to present the theory and practice of the techniques in this course. Although the lectures will primarily focus on the key concepts of software engineering, some lectures will also include practical demonstrations of various key technologies required for the implementation of the group project. Lecture slides will be available on the course web page.

Tutorials

Tutorials help clarify ideas from lectures and work through exercises based on the lecture material. You should make sure that you use them effectively by examining in advance the material to be covered in each week's tutorial, by asking questions, by offering suggestions and by generally participating. The tutorial questions will be posted on the course website in the week before each tutorial.

Laboratories

Laboratories are two-hour sessions in a computer lab and focus either on:

1. Reviewing the end of a milestone for the group project (Weeks 4, 7, 10)
2. Completing small practical exercises involving tools/frameworks from lectures (Weeks 1-3, 5-6, 8-9).

For all labs concerning (2), the exercises will be released in the week preceding the lab class. **Completed solutions must be submitted by the expected due-date at the end of that week.** Your tutor will not award any marks for the lab exercise if you have not submitted a complete solution. Marks will only be awarded for that lab when your tutor manually marks you off for it (after checking it) when you attend the lab in the proceeding week it's due. E.G. Your tutor will mark off your lab for week 4 in week 4 or week 5.

You **cannot** obtain marks by e-mailing lab work to tutors. All labs must **normally** be submitted on time and, if necessary, demonstrated. Your tutor may grant you an extension on the lab exercise if you have been unable to complete the lab exercise due to illness or unavoidable circumstances (e.g, jury duty).

At the beginning of each lab, as of week 2, your tutor will do a "check-in" with you to see how you're going with the project at the beginning of each lab. This check in will influence your tutors interpretation of your team's diligence and time management, which will inevitably influence your final mark.

Submission of lab practical exercises (2) will contribute to **14%** of your overall course mark. Each lab is worth 2%.

Group Project

There will one main group project which will run through the teaching period from weeks 2-10 and contributes to **36%** of the overall course mark. The specification of the group project will be released at the end of week 1.

You are required to form groups of 4 (or 5 if tutorial sizes require it and your tutor approves it) for the group project. These groups must be within your own tutorial group.

The group project will be implemented using an agile software delivery mode. As such, your team will be required to build and deliver the project in iterations (milestones). Each iteration will deliver a part of the requirements of the project and will encompass all the SDLC activities, namely analysis, design, coding and testing. At the end of the iteration, you (as a team) will demonstrate to your lab class the functionality implemented during that iteration cycle. Changes to project requirements are a natural and unavoidable part of any software project life-cycle. Hence, students will need to bear in mind that project requirements may be subject to change and enhancements to functionalities may be made at the end of each iteration. You will need to carefully design the solution for your current iteration, such that the solution is extensible to accommodate these changes.

The deliverable for each iteration (source code, design artifacts etc) will usually be due at the start of the week the iteration is due and each iteration must be demonstrated (by the entire team) on the scheduled date. Late demonstrations will not be accepted. **All team members must be present for the iteration demo.** If any team member cannot be present, they must notify the lecturer and tutor prior with a valid reason. A mark will be awarded at the end of each iteration, which will go towards the overall mark for the group project. The final demo of the group project will be conducted in week 10. Please note, as there are two public holidays in week 10, if your lab session falls on one of these days, your project demo will be in week 11 (as advised in the class time-table). The mark for iteration demonstration of the group project will go towards the mark allocated for your group project.

Tutors will continually monitor the GitLab repositories to see the team's progress and individual member's contribution to the group project.

Final Exam

There will be a centrally timetabled final exam which will in your UNSW exam timetable. The exam will contain a mixture of multiple choice questions, "theory" and design questions and programming exercises. More specific details of the exam will be provided through the course.

There is a hurdle requirement on the final exam. If you do not score an adjusted mark of at least 50% in the final exam, you cannot pass this course. If your overall course score exceeds 50%, but your exam performance is not satisfactory (<50%), the hurdle will be enforced via a grade of UF. Of course, if your overall course score is less than 50%, then your grade will be FL.

If you cannot attend the final exam because of illness or misadventure, then you must submit a Special Consideration request, with documentation, through MyUNSW within 72 hours of the start of the exam exam. If your request is reasonable, then you will be awarded a Supplementary Exam. No supplementary exams will be provided for students who score marks 49 or below on grounds of being "close" to a pass.

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Plagiarism is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

Assessment

Item	Due	Marks
Lab Activities	Weeks 1,2,3,5,6,7,9	14%
Major Project	Week 10, with interim milestones Week 4, Week 8	36%
Final Exam	Exam period	50%

Course Schedule

Week	Lectures	Assessment / Lab Item
1	• Introduction • Testing • Teamwork / soft skills • Git • Project Intro • Python programming language	Lab 1 Due
2	• Requirements analysis • User stories • Project methodologies (agile) • Python programming language	Lab 2 Due
3	• Use cases • Software complexity • System schematics • Python programming language	Lab 3 Due
4	• Web Protocol / HTTP • Flask • App State • Web Authentication	Project Milestone 1 Due
5	• Testing • Software processes • Measurements and optimisation	Lab 5 Due
6	• Web browser & Front-end programming • HTML/CSS • Javascript • ReactJS	Lab 6 Due
7	• Json/XML/Yaml • Persistence of data • Python programming language	Project Milestone 2 Due
8	• Conceptual modelling • Software design • Effective use of libraries • Python programming language	Lab 8 Due
9	• Deployment • Python programming language	Lab 9 Due

10	• Exam • Revision	Project Milestone 3 Due
11	No lectures	
12	• Revision Lecture (Keith Burrows Theatre)	

Tutorials will *typically* cover content from the lectures of the week previous. E.G. Tutorials in week 3 will be regarding content in week 2

Resources for Students

There is no single text book that covers all of the material in this course at the right level of detail and using the same technology base as we are. The lectures should provide sufficient detail to introduce topics, and you will then study them in further depth in the tutorials, labs and group project. For some lectures, further reading material may be given for students who wish to gain a deeper understanding.

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

However, during the term students are encouraged to provide feedback both during lectures, during tutorials, and generally to course staff via email. Anonymous feedback forms will be provided to students throughout the course.

In the previous offering of this course, issues that students have pointed out include:

- Too much content in the course for it's duration
- Not enough cohesion between lectures and laboratories

This have been addressed during the 19T3 offering.

Resource created 8 months ago (Wednesday 28 August 2019, 09:57:54 PM), last modified 5 months ago (Wednesday 13 November 2019, 02:06:44 PM).

Comments

(/COMP1531/19T3/forums/search?forum_choice=resource/31863)

(/COMP1531/19T3/forums/resource/31863)

There are no comments yet.