Resources / Course Outline

Course Outline

Contents

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

1. Course Details

Course Code	COMP2511
Course Title	Object-Oriented Design & Programming
Convenor	Dr Ashesh Mahidadia <ashesh@cse.unsw.edu.au>
Admin Team	Course Account Email <cs2511@cse.unsw.edu.au> **Unless you specifically require to contact a member of the admin team,** **please use the above email for any queries related to the course** Braedon Wooding <b.wooding@unsw.edu.au>, Nick Patrikeos <n.patrikeos@unsw.edu.au>, Matthew Perry <matthew.perry@unsw.edu.au>
Classes	Timetable for all classes (http://timetable.unsw.edu.au/2021/COMP2511.html)
Consultations	To be added later
Units of Credit	6
Course Website	COMP2511 Class Webpage (https://webcms3.cse.unsw.edu.au/COMP2511/21T2/) (https://webcms3.cse.unsw.edu.au/COMP2511/21T2/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2511.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2511.html)

2. Course Summary

COMP 2511 covers the theory and practice of object-oriented design and programming with an emphasis on teaching students how to apply software design principles and design patterns to the building of flexible, reusable and maintainable systems. The course also introduces user-interface design and programming and includes the implementation of a large project done in pairs using agile software practices and enables students to have hands-on experience in applying the principles taught, implementing and incorporating the use of good design principles and design patterns in software programs.

2.1 Assumed Knowledge

Students are assumed to be:

- competent C programmers who can understand and use abstract data types
- understand and know how to use git reasonably
- be familiar with fundamental object-oriented design concepts

This course expects that students have undertaken COMP1531 (a prerequisite for COMP 2511 introduced following a curriculum review 3 years ago) prior to taking this course. COMP1531 introduces use of git and also exposes students to the fundamental principles of software design. However, there are a small number of students on the old program who have not taken COMP1531. For these students, the week 1 lab is intended to bring them up to speed with git.

2.2 Student Learning Outcomes

The following are the intended learning outcomes of this course:

- *Object-Oriented Design*
  - Understand the principles of object-oriented design
  - Be able to follow a systematic object-oriented design process
  - Be able to interpret and use tools for object-oriented design
  - Learn how to apply design principles and design patterns effectively to design flexible, maintainable and reusable systems
- *Object-Oriented Programming and Java*
  - Understand object-oriented programming languages
  - Be able to write medium-scale object-oriented programs in Java
  - Apply systematic methods such as test driven development
  - Understand the importance of documentation and use tools to document software
  - Be able to use an IDE (Interactive Development Environment) for software development
- *Software Processes*
  - Understand the importance of team organization and communication
  - Be able to work within a small team in the context of a software development project
  - Be able to plan and execute a software project according a systematic software process
- *User Interfaces*
  - Become familiar with principles of effective user interface design
  - Be able to implement a user interface in Java

This course contributes to the development of the following graduate capabilities:

This course contributes to the following UNSW graduate attributes.

- *An in-depth engagement with the relevant disciplinary knowledge in its interdisciplinary context.* The development of systems suitable for different user groups encourages consideration of the users of technology.
- *The capacity for analytical and critical thinking and for creative problem solving.* The creation of software applications requires creative approaches to design and implementation and applying best practices.

- *The ability to engage in independent and reflective learning.* Software design and implementation must be done independently and students benefit by reflecting on their experiences.
- *The skills required for collaborative and multidisciplinary work.* Through a small team project, students gain experience in working within a collaborative environment.
- *The skills of effective communication.* Through a small team project, students develop interpersonal communication skills and presentation and writing skills.

3. Teaching Strategies and Rationale

There are some key changes to the structure and content of the course this term. These are strategic changes necessary to improve the student learning experience and outcomes and to adapt to the COVID 19 situation.

3.1 Lectures (4 hours each week)

Lectures will be used to present key concepts and practical examples of design techniques taught in this course. Lectures will be delivered on *Blackboard Collaborate,* during weeks 1 to 5, and 7-10.

3.2 Tutorials

A 1 hour tutorial session per week, which is scheduled before the lab. Tutorials will serve three purposes (1) clarify ideas from lectures and work through exercises based on the lecture material (2) provide practical demonstrations in the Java programming language (3) peer code review. 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.

Tutorials will often involve collaborative and break-out work where you will work with a group of other students to solve or answer problems. This will often be done in your major project group.

Tutorials contribute to your class mark, see the section " *Class Mark (10%)* " below.

The tutorial questions will be posted on the Web in the week before each tutorial, and the solutions will be available at the end of the corresponding week.

Tutorials/Labs will be run via ***MS Teams*** . MS Teams is chosen in preference to Blackboard Collaborate due to the relatively higher degree of reliability it provides, the ability to remote control if necessary, and that the features of of Blackboard Collaborate that are useful in other courses (e.g. COMP1511) are not needed in the structure of how COMP2511 is operated.

All tutorials start in week 1.

3.3 Laboratory classes

Following the tutorial class each week, there will be a two-hour laboratory class, during which you will work on a variety of small practical problems that will enable you to apply the concepts taught in the lectures.

Each week, there will be a few exercises to work on. These exercises will be released in the week preceding the lab class. The exercises will need to be submitted and will be assessed by your tutor. This term, lab exercises will constitute an important part of the course assessment, see the section " *Class Mark (10%)* " below.

You should normally get your lab work assessed during the week for which it is scheduled (i.e. you must complete the week 1 lab exercise during the week 1 lab). All completed lab exercises must be pushed to GitLab and submitted prior to being marked by your tutor or lab assistant. If you are unable to finish the exercises during the lab, you may continue working on it during the week, but **you must submit the lab by the following Sunday 5:00 pm** in order to get any marks for it. You must then demonstrate your work to your tutor during the following week's lab. The code that you submit by Sunday is what will be assessed. You must not make any changes to the completed solution after Sunday. Your tutor will check the time-stamp prior to marking the lab.

Summary: to obtain any lab marks for the Week X lab, you must do 2 things:

- *submit your lab work by the following Sunday 5:00pm*
- *demonstrate your work to your tutor in the week X lab class*
  OR *demonstrate your work at the start of the lab in week X+1*

You **cannot** obtain marks by e-mailing lab work to tutors. Your tutor will provide feedback during a lab session on your approach to the problem and on the style of your solution. All labs must **normally** be demonstrated OR submitted by the due-date.

Periodically, the lab sessions will also be used to schedule demonstrations of your ongoing group project. As the implementation of the group project will be based on an **agile software development methodology** , you will required to demonstrate progress of your project in iterations. No excuses will be accepted (unless unavoidable circumstances) for any member of the group not being present for the iteration demo.

As this course has a significant practical component, laboratory classes are important to help you acquire the necessary skills in the relevant tools and frameworks that will be used in the implementation of the group project and also prepare you for the final exam. If you do not put a good amount of effort into the lab classes you risk failing the group project and the final exam.

Use of git and GitLab will be strongly enforced in this course, as using them for the medium of collaboration and source control is a key learning outcome that will be applied by students in several following courses through their degree.

All labs start in week 1.

3.4 Help Sessions

Help sessions are unprepared drop-in "clinics" where students and groups can go to seek help about course related matters, whether that be the project, tutorials, or labs. Current tutors or lab assistants will supervise each help session.

Help sessions will be run via Blackboard Collaborate.

3.5 Assignment and Project

There will be one **assignment** (a design-based) to be completed **individually** which will be released at the end of week 1 and will be due on Tuesday of Week 4 . This assignment will constitute 15% of the overall course mark.

There will be one main **project** (**group of 4 students**) which will run through the rest of the teaching period and contributes to 35% of the overall course mark. You are required to **form your group (of 4)** for the project **by the end of Week 1** . Importantly, all group members must be **from the same tutorial/lab group** . The specification and other details of the group project will be released in Week 3.

3.6 Final Exam

There will be a centrally timetabled final exam which will in your UNSW exam timetable. There is a hurdle requirement on the final exam. If you do not score at least 50% on the exam, you cannot pass this course. If your overall course score exceeds 50%, despite scoring very poorly (<50%) on the exam, 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.

3.7 Tools Used in Assessments

Use of git and GitLab will be strongly enforced in this course, as using them for the medium of collaboration and source control is a key learning outcome that will be applied by students in several following courses through their degree.

All work for the assignment, project, and labs will only be considered for assessment if pushed/uploaded to the platform specified in the assessment specification. This includes that only work pushed/uploaded to the correct platform will be considered for determining contribution levels; and the owner of the account pushing/uploading

the work will be deemed the sole author of the work (so you should not allow your colleague to push your work).

4. 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 you are using these facilities. 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)

5. Assessment

Item	Due	Marks
Assignment	Week 04 (Tuesday)	15%
Project	Milestone submissions: Weeks 5 (Friday), 8 (Monday), 10 (Monday)	35%
Class Mark (tut/lab mark)	All Weeks	10%
Final Exam	Exam period	40%

5.1 Class Mark (10%)

Your class mark is made up of marks associated with tutorials, and marks associated with labs. There is a total of 24 marks that can be gained between 9 tutorials and 8 labs throughout the course (i.e. 4 bonus marks), although it will be capped at 20 overall and comprise 10% of your overall course mark.

- You will receive marks out of 2 for each lab. A rough guideline for how marks are awarded for your **lab** is below:
  - *2 / 2 marks (completion of all required tasks)*
  - *1.5 / 2 marks (satisfactory completion most tasks)*
  - *1.0 / 2 marks (did an OK job overall, satisfactory with some errors or items missing)*
  - *0.5 / 2 marks (completed some things but did so quite poorly)*
  - *0.25 / 2 marks (barely an attempt)*
- You will receive mark out of one for each tutorial. A number of factors contribute toward you receiving a high grade in this area, including but not limited to:
  - *Attending the full tutorial*
  - *Asking good and logical questions throughout the tutorial*
  - *Taking initiative to answer questions and be engaged*
  - *Attending with your webcam on for online tutorials (or your phone if you don't have)*

You will be notified by the course authority in week 5 as to how your tutorial attendance and participation mark is progressing.

This marking structure is designed so that students can make choices in how they want to learn. Some students may want to spend time attacking lab exercises, while others want to spend time engaging more in tutorials. With 4 extra bonus marks there is plenty of wiggle room to gain the marks.

5.2 Assignment (15%)

The assignment is to be completed **individually.** The assignment will be released at the end of week 1 and will be due at 8am Tuesday of Week 4. The assignment will constitute 15% of the overall course mark.

5.3 Project (35%)

The specification of the group project will be released in Week 3. You are required to **form groups of 4** for the group project **by the end of Week 1** . Importantly, all group members must be **from the same tutorial/lab group** . Details about the group project will be released in Week-3. The project will constitute 35% of the overall course mark. There are three milestones in the project. You need to submit milestone 1 in week 5 and

provide demo in week 7 for assessment. You need to submit milestone 2 on Monday of week 8 and provide demo in week 8 for assessment. You need to submit milestone 3 on Monday of week 10 and provide demo of the entire project in week 10 (final demo) for assessment.

5.4 Final Exam (40%)

Your final mark for this course will be computed using the above assessments as follows:

- Course_Work_Mark = lab_mark + assignment_mark + project_mark (out of 60)
- Exam_Mark = Final Exam Mark (out of 40)
- Exam_OK = Exam_Mark >= 20
- Final_Mark = Exam_Mark + Course_Work_Mark
- Final_Grade =UF, if !Exam_OK

Students are eligible for a Supplementary Exam if and only if:

- students cannot attend the final exam due to illness or misadventure. Students must formally apply for a special consideration, and it must be approved by the respective authority.

6. Course Schedule

Please note the following schedule is *subject to change* .

Week	Topic	Assessments
1	Course Introduction, Introduction to Java (JDK, J2EE) What is OO design? Object, Classes, Constructors	-
2	Abstraction, Encapsulation and Inheritance, OO Relationships, Exceptions in Java, etc.	-
3	Interfaces, Design Principles - Law of Demeter, LSP, Introduction to Refactoring	-
4	Design Principles, Strategy Pattern, State Pattern, Introduction to the Project, Agile Software Development and the Project	* Assignment Due (Tuesday)
5	Observer Pattern, Composite Pattern, UI/UX Fundamentals JavaFx	* Project - *Submission* of Milestone 1
6	-	-
7	Design By Contract Pattern, Generics, Collections in Java; Iterator Pattern,	* Project - *Demo/Assessment* of Milestone 1

8	Refactoring, Code Smell, Decorator and Adapter Patterns, Template Pattern	* Project - Submission of Milestone 2 on Monday, and *Demo/Assessment* of Milestone 2 during your lab time.
9	Creational Patterns (Factory Method, Abstract Factory, Builder, Singleton Patterns) Visitor Pattern	-
10	The Future of Object-Oriented Designing, Course Revision and Exam Structure	* Project - Submission of Milestone 3 on Monday, and *Final Demo/Assessment* during your lab time.

7. Resources for Students

There is no single text book that covers all of the material in this course at the right level of detail. The lectures should provide sufficient detail to introduce topics, and you will then study them in further depth in the labs and group project.

There are also many online resources available, and we will provide links to the most useful ones. Some are listed below. If you find others, please post links in the Comments section on the Course Outline page.

Some suggestions for books that cover at least some of the topics in this course

- *Head First Design Patterns* , by Elisabeth Freeman and Kathy Sierra, The State University of New Jersey
- *Refactoring: Improving the design of existing code* , by Martin Fowler

8. 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.

The main improvements to be made in 21T2 include developing new assignment and project closely aligned with the course learning objectives, strongly encouraging group work during the project, closely aligning the lecture content with the tutorials/labs, and making the tutorials more collaborative in nature.

Resource created 6 months ago (Sunday 23 May 2021, 09:34:52 AM), last modified 4 months ago (Sunday 18 July 2021, 05:03:20 PM).

Comments

(/COMP2511/21T2/forums/search?forum_choice=resource/61325)

(/COMP2511/21T2/forums/resource/61325)

There are no comments yet.