Course Outline

Contents

Course Details

Course Code	COMP2511
Course Title	Object-Oriented Design & Programming
Convenor	Dr Ashesh Mahidadia (ashesh@cse.unsw.edu.au)
Admin	Matthew Perry (matthew.perry@unsw.edu.au)
Classes	Timetable for all classes (http://timetable.unsw.edu.au/2020/COMP2511.html)
Consultations	To be added later
Units of Credit	6
Course Website	COMP2511 Class Webpage (https://webcms3.cse.unsw.edu.au/COMP2511/20T3/) (https://webcms3.cse.unsw.edu.au/COMP2511/20T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2511.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2511.html)

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.

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.

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.

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.

The revised structure is as follows:

Lectures - 4 hours each week

A total of 36 (9x4) hours of lectures will be used to present key concepts and practical examples of design techniques taught in this course.

Tutorial recording - 1 hour each week

You will be expected to watch a 1-hour tutorial recording prior to attending your scheduled tutorial Q&A session each week. This video will discuss questions from that week's tutorial content.

Tutorials Q&A - 1 hour each week

You will be registered to a scheduled tutorial session via myUNSW. Your tutor or lab assistant will be available during your tutorial time-slot to answer any questions you may have regarding the tutorial questions after watching the tutorial recording.

Laboratory classes (contribute to 10% of overall course mark)

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.

There are six labs (weeks 1, 2, 3, 5, 7 and 9) with specified lab exercises. Your tutor or lab assistant will be available during your lab time-slot to answer any questions you may have regarding the lab questions. The other labs will be used for lab and project assessments. Please note that you must attend all nine lab and tutorial sessions during the term. Lab exercises will be released in the week preceding the lab class. The lab exercises will need to be submitted and will be assessed by your tutor. Lab exercises constitute an important part of the course assessment and will contribute to 10% of the overall course mark.

You should **demonstrate** your work to your tutor during the lab or the following week's lab for feedback on your problem solving, approach, the style of your solution and importantly **marking** .

Summary: you must,

- *demonstrate your work to your tutor in the week X lab class* (for feedback and marking)
- OR *demonstrate your work at the start of the lab in week X+1* (for feedback and marking)

The questions will be related to (and may not the same as) the lab questions. Please note that you must briefly justify your answers. Based on the quality of your justifications and answers, your tutor will award you marks. You don't need to offer long answers, as far as you convey your logic correctly and your tutor understands it, that's fine.

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.

All labs start in week 1.

Assignment and Project

There will be one assignment (design-based) to be completed individually which will be released in week 2 and due on Sunday of Week 4. This assignment will constitute **15** % of the overall course mark.

There will be one main project (group of 2 students) which will run through the rest of the teaching period and contributes to **35%** of the overall course mark. The specification of the group project will be released in week 3. You are required to form groups of 2 for the group project. Details about the group project will be released through the term.

Periodically, the tutorial and 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 be required to demonstrate progress of your project in iterations. In these sessions, similarly to the lab assessments, students/teams will be asked questions about the assignment/project. These sessions will be in a time-slot specified by the tutor, and must be attended and answered both verbally and satisfactorily to receive marks.

Students missing the demonstration will automatically receive 0 for that iteration unless special consideration is approved - no exceptions will be made to this policy. Please note that these interactive sessions will be conducted in Microsoft Teams or Blackboard Collaborate, and they will be recorded for future reference.

You must be setup with any necessary software for demo sessions open and running at the commencement of the session. If avoidable technical issues occur (e.g. because you haven't opened the required windows/software or haven't run the required commands) no additional time will be provided. In such a circumstance, the unanswered questions will be assigned 0 marks.

The level of contribution by each team member will be analysed at the end of the trimester, and positive and negative scaling may be applied in cases of uneven contribution.

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.

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).

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)

Assessment

Item	Due	Marks
Assignment	Week 04 (Sunday)	15%
Project	Milestones: Weeks 5,7,9	35%
Labs	All Weeks	10%
Final Exam	Exam period	40%

Final Mark

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.

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	-
3	Interfaces, Design Principles - Law of Demeter, LSP, Introduction to Refactoring	-
4	Design Principles, Strategy Pattern, State Pattern	Assignment Due (Sunday)

5	Observer Pattern, Composite Pattern, JavaFx	Project - Milestone 1 Submission
6	-	-
7	Design By Contract Pattern, Generics, Collections in Java; Iterator Pattern,	* Project - Milestone 1 Demo/Assessment Project - Milestone 2 Submission
8	Decorator and Adapter Patterns, Refactoring, Code Smell, Template Pattern, User Centred Design and MVC	Project - Milestone 2 Demo/Assessment
9	Creational Patterns (Factory Method, Abstract Factory, Builder, Singleton Patterns) Visitor Pattern	Project - Milestone 3 Submission
10	Course Revision and Exam Structure	Project - Final Demo

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

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

Resource created a day ago (Thursday 10 September 2020, 10:01:34 PM), last modified about 20 hours ago (Friday 11 September 2020, 01:41:57 PM).

Comments

(/COMP2511/20T3/forums/search?forum_choice=resource/53033) (/COMP2511/20T3/forums/resource/53033)

Add a comment

There are no comments yet.