Math 2415 Calculus I
Fall 2019 Wenzhou-Kean University

I. Course Sections and Instructor Information

Section: W07 Meeting Times: 04:00-05:45 PM Days: TF Location: GEH B509
Instructor's Name: Dow Inouye E-mail address: dinouye@kean.edu
inouyedk@wku.edu.cn
Office: GEH B417 Office Phone: 5587-0454
Office Hours: **M** 10:00-10:45 AM, 2:45-4:00 PM, 6:30-7:30 PM; **T** 1:45-3:45 PM, 6:30-7:30 PM
Th 10:00-10:45 AM, 2:45-4:00 PM, 6:30-7:30 PM; **F** 1:45-3:45 PM, 6:30-7:30 PM

Prerequisites: Math 1054 or placement

Textbook: *Calculus: Early Transcendentals*
Second Global Edition
Author: Briggs, W., Cochran, L., Gillett, B. ISBN: 10: 1-292-06231-2

II. Course Description

This course includes the following topics: Limits, Rates of Change and Tangent Lines, Continuous Functions, Trigonometric Limits, Intermediate Value Theorem, Derivatives, the Chain Rule, Implicit Differentiation, Extreme Values, Graph Sketching, Optimization Problems, Definite Integrals and the Substitution Method.

III. Course Objectives

Students of Math 2415 will strive to demonstrate their understanding of limits for single variable functions, differentiate and integrate algebraic and trigonometric functions of a single variable, use the derivative concept to analyze the behavior of single variable functions and their graphs, and apply technology to demonstrate the use of calculus.

IV. Course Requirements

A. Exams

There will be three in-class exams. All exams are **closed book and closed notes** and will be approximately 90 minutes in length. These exams will be announced at least two meetings in advance. (See the Schedule of Lessons, which gives the approximate dates.) **All exams are to be written in pencil**, and not in pen, so it is highly recommended to invest in a high-quality mechanical pencil and eraser. There will be no draft or extra paper allowed on exams.

B. Quizzes and Homework Assignments

There will be approximately five (5) announced in-class quizzes taken during the beginning of the class period. **Quizzes will be written in pencil**, not in pen. No draft or extra paper will be allowed on quizzes.

Problems from the textbook will be assigned on a regular basis. A select few of these problems will be turned in separately (usually at the next class meeting) and will be checked for completeness, accuracy, proper notation, organization, etc., but will not be graded with a score. Instead, these assignments will be recorded as credit, half-credit, or no credit. **Turned in assignments are to be done on A4-sized paper (lined or unlined)**. Assignments may be done in pen if the work is presented neatly and conscientiously.

C. *Attendance Policy*

Your instructor may record class attendance with an attendance sheet.

D. *Make-Up Policy*

1. **Late homework assignments** will be accepted at the instructor's discretion. Assignments turned in after the end of class time—including due to an excused absence—will be considered late (and may be subject to penalty).

2. **Missed quizzes** may be made up by prior arrangement if a student's absence is due to an academic or professional-related engagement. No quiz scores will be dropped, so it is imperative that you take every quiz as scheduled. Quizzes missed due to serious illness will be handled on a case-by-case basis.

3. **Missed exams** can be made up within one week of the actual in-class examination with a valid reason. However, you must contact your instructor as soon as possible to arrange the scheduling of the make-up exam within 48 hours of the date of the exam that you missed. If you are unable to contact your instructor, please have a reliable person do this on your behalf. Valid documentation may be presented to your instructor. Examples of valid documentation include a doctor's note for an illness, obituary or funeral announcement for a death in the immediate family, or a copy of court documents for a legal summons. In any event, contact your instructor immediately, as missing an exam without immediate notification is a serious academic matter.

E. *Method of Evaluation/Assessment*

Three (equally weighted) Exams	66%
Quizzes	22%
Homework Assignments	12%

F. *Grading Policy*

You can contact your instructor if you have questions about your grade during the semester. However, because of the weighting detailed above, your grade is fluid throughout the semester, so only an estimate can be given at any time. From past experience, your quiz scores are an accurate reflection of how you are doing in the course.

Your grade will be calculated based on the following guidelines:

Percentage:

93 and above	A	75-79	C+
90-92	A-	70-74	C
87-89	B+	60-69	D
84-86	B	< 60	F
80-83	B-		

While being respectful of the above grading scale, grades will ultimately be determined by analyzing a frequency distribution of the class scores, with borderline grades being determined by a holistic evaluation of a student's performance throughout the semester (which can include attendance, class participation/ attentiveness, completion of assignments, any academic integrity issues, etc.).

Academic Early Alert/Mid-Term Progress information will be provided to students by the instructor via KeanWise. Please see the 2019-2020 Undergraduate Catalog for more information.

G. ***Use of Electronic Devices during Quizzes and Exams***

Cell phones, laptops, or other electronic devices are prohibited during quizzes and exams and should be stored away out of sight, not merely turned off.

H. ***Instructional Methods***

This course is taught by discussing definitions, working sample problems, proving theorems, and through the assignment of daily exercise problems.

Open Educational Resources

For more information, please see the Nancy Thompson Learning Commons:
http://libguides.kean.edu/OER

V. Important Dates

Last Day to Add/Drop Courses: September 09, 2019 (Monday)
Last Day to Withdraw from Courses with a "W" Grade: October 21, 2019 (Monday)

See http://www.wku.edu.cn/en/academics/academic-calendar/ for other important dates, or inquire with the Registrar's Office.

VI. Each student is responsible for reading and understanding the contents of the

University Academic Integrity Policy
http://www.kean.edu/media/academic-integrity-policy;

Student Code of Conduct, as it discusses expectations of appropriate conduct in the classroom:

http://www.wku.edu.cn/en/org/student-affairs/code-of-conduct/.

All students must have and use a valid **WKU email account**. For those who do not already have one, please contact the Registrar's Office. (You will also need a valid **Kean email account**, as group mailings via *Blackboard* will be sent to your Kean address.)

VII. Tentative Schedule of Lessons*

** Your instructor reserves the right to amend this schedule (and syllabus) as deemed necessary.*

Date	Sections	Date	Sections
9/03	Sec. 2.2	11/01	Sec. 4.2, 3.2
9/06	Sec. 2.6, 2.3	11/05	Sec. 4.4
9/10	Sec. 2.4, 2.5	11/08	Sec. 4.5
9/17	Sec. 2.7, 2.1, 2.6	11/12	Review for Exam 2
9/20	Sec. 3.1, 3.2	11/15	Exam 2
9/24	Sec. 3.3, 3.4	11/19	Sec. 4.6
9/27	Sec. 3.5	11/22	Sec. 4.9
9/28	Sec. 3.7 (Saturday)	11/26	Sec. 5.1, 3.2, 4.2
10/08	Review for Exam 1	11/29	Sec. 5.2
10/11	Exam 1	12/03	Sec. 5.3
10/15	Sec. 3.6, 3.8	12/06	Sec. 5.3, 5.4
10/18	Sec. 3.9	12/10	Sec. 5.5, 6.8
10/22	Sec. 1.4, 3.10	12/13	Sec. 5.5, 6.2
10/25	Sec. 3.11	12/17	Review for Exam 3
10/29	Sec. 4.1	12/20	Exam 3

"The roots of education are bitter, but the fruit is sweet."
~ Aristotle

Here is a template for addressing turned in homework exercises.

Top of Page (21 cm)

Assignment #___

Family Name Given Name
English Name (if any)
MATH 2415-W07

- Assignments with no name shall receive a "0" and will not be returned.
- Papers with no assignment number shall receive at most half-credit.
- Assignments disregarding the above format shall receive at most half-credit.
- Assignments requiring multiple sheets must be stapled (in the upper-left corner).

N.B. For turned-in exercises and especially on quizzes and exams, **you must show the process by which you arrived at your answer in order to receive full credit**. Do not erase any supporting work, no matter how trivial you might think it is. Willfully erasing supporting work will be construed as engaging in academic deception or even as colluding/participating in academic dishonesty.

VIII. Disabilities & Non-Discrimination Statement

Wenzhou-Kean University is an affirmative action, equal opportunity institution. Students with documented disabilities who may need special instructional accommodations or who may need special arrangements in the event of an evacuation should notify the instructor as soon as possible, no later than the second week of the term. Students may contact Ms. Lin TENG at the Office of the Vice Chancellor for Student Affairs, General Education Building, A215,
Email: vcsa@wku.edu.cn.

IX. Title IX

Title IX of the Education Amendments of 1972 (Title IX) prohibit discrimination on the basis of sex in education programs or activities. Sexual harassment in any form will not be tolerated at Kean University. Sexual harassment by students should be reported to the Office of Affirmative Action Programs, Office of the Vice President for Student Affairs or the Office of Community Standards and Student Conduct immediately. Information about the University's Sexual Misconduct Policy may be found at the following: http://www.kean.edu/policies/sexual-misconduct-policy. Students may contact Dr. Shuli XU (shulix@wku.edu.cn), the Title IX contact person at WKU.

X. KU Non-Discrimination Policy:

Kean University is an affirmative action, equal opportunity institution.

Math 2995 (Linear Algebra I) Spring 2020

I. Course and Instructor Information

Section	01 W 8:00 -10:45 am (B206)
Instructor:	Dr. Pablo Zafra
Telephone number:	908 737 3709
Email address:	pzafra@kean.edu
Office:	C213
Office Hours:	1/21 – 4/22: TTh 7:45 – 9:30 am, 10:45 am – 12:30 pm W 7:30 – 8:00 am, 10:45 – 11:30 am F 8:45 AM - 12:30 pm (by appointment only) 4/23 – 5/6: TTh 7:30 – 9:30 am, 10:45 am – 12:30 pm W 7:30 – 8:00 am 5/7 – 5/13: TTh 8:15 – 9:30 am, 10:45 am – 12:30 pm

II. Course Description

Systems of linear equations, matrices, determinants, vector spaces, linear transformations, eigenvalues and eigenvectors, and applications.

Corequisite: Math 2415 or the equivalent

III. Course Objectives

Upon completion of this course, the student should be able to:

- Understand the basic concepts of matrix and linear algebra
- Solve systems of linear equations utilizing matrix techniques
- Solve problems involving various applications of linear algebra
- Use the computer (and/or calculator) to facilitate the solution of problems involving matrices

IV. Textbook & Materials

Textbook Title: Linear Algebra and Its Applications, Fifth Edition by David C. Lay, et. al.

Publisher: Pearson, 2016

ISBN-13: 9780134022697 (bundled with MyLab access card)

Note: As an alternative, a MyLab Math (MyMathlab) access code which includes the ebook may be purchased separately. ISBN-13: 9780321989871 www.mymathlab.com

Calculator: TI-84+ or comparable

V. Course Requirements and Evaluations

A. ATTENDANCE/CLASS PARTICIPATION

Regular attendance is crucial to your success in this course. Although class attendance will only be taken occasionally, you are expected to be present and be on-time to every class meeting. Attendance does not contribute a specific percentage towards the final grade, but as an incentive for good attendance record and active class participation, borderline cases are generally considered for the higher letter grade.

Note that you will be responsible to catch up and study any missed lecture on your own. You are also responsible to finish on-time the online homework exercises (see below) that are related to the missed class. Therefore, do your best and attend every class (or find a classmate to exchange contact information in case you missed a class so you can borrow class notes.)

B. HOMEWORK

To supplement and reinforce concepts discussed from the lecture, you will be required to work on a series of online exercises from MyMathLab that are similar to selected textbook exercises (more details to follow). You are strongly encouraged to work on all the assigned exercises if you want to do well on the in-class exams. You may want to take advantage of the free tutoring in room C-242 Scicence Bldg for help in the homework assignments. **Grades from these online assignments will contribute 20% to your final grade.**

C. EXAMS

There will be three in-class exams tentatively scheduled on week 5, 11 and 17 of the semester, respectively. Alternative arrangements for missed exam will be permitted only under exceptional circumstances, and should be taken within one week of the actual in-class examination. Therefore, you are strongly advised to notify me as soon as you know that you will not be able to take the scheduled exam, and to contact me no later than the day after the exam if you missed it. If I am not notified of your absence within this timeframe, I will assume that you do not have any intention to take a make-up exam and a grade of 0 will be recorded.

Note that only one missed exam (with valid documentation) can be made-up during the semester. **Exams 1 and 2 will contribute 25% each towards the final grade. The final exam contributes 30% to your final grade.**

Please do not ask for extra credit or curves on any exams.

D. APPROXIMATE GRADING SCALE (based on overall total percentage)

93-100 **A** 89-92 **A-** 86-88 **B+** 82-85 **B** 78-81 **B-** 72-77 **C+** 66-71 **C**
60-65 **D** 0-59 **F**

VI. Important Dates

January 21	Term begins
January 28	Last day to withdraw with 100% refund
February 4	Last day to withdraw with 75% refund
February 11	Last day to withdraw with 50% refund
February 17	President's Day – University closed/No classes
March 9-15	Spring Recess
April 9	Last day to withdraw from courses with a **W** grade, 0% refund
May 5	Classes follow Friday schedule; No Tuesday classes meet
May 6	Classes follow Monday schedule; No Wednesday classes meet
May 13	Term Ends

VII. Important University Policies and Information

- Students are responsible to review and understand the *University Academic Integrity Policy* (available at the Center for Academic Success or athttp://www.kean.edu/admin/uploads/pdf/AcademicIntegrityPolicy.pdf)

- Students should review the *Student Code of Conduct*, as it discusses expectations of appropriate conduct in the classroom: http://www.kean.edu/KU/Code-of-Conduct.

- The **Students Rights and Responsibilities handbook** is available at:
  http://www.kean.edu/KU/Forms-Policies-and-Publications.

You are expected to conduct yourself in a manner that is consistent with the learning mission of the University. All forms of academic dishonesty are strictly forbidden. This includes but is not limited to the following: communicating with other students during exams; unapproved references to books, notes or "cheat sheets" during exams; and plagiarism—representing another person's work as your own. You should be aware that plagiarism is generally easy to recognize. In instances where I suspect plagiarism, I reserve the right to ask you to orally defend your work. The minimum penalty for an incident of academic dishonesty will be failure of the assignment where the dishonesty occurred. Do not underestimate my determination to enforce this policy.

For further information on academic ethics, please consult the Student Handbook found on the weblink above.

- Students are strongly encouraged to register for the University's emergency notification system (www.mir3.com/kean) in order to be informed of campus emergencies, weather notices, and other announcements.

- All students must have a valid Kean email account. For those who do not already have one, forms are available on-line at http://www.kean.edu/KU/Forms-OCIS; click on E-mail Account Request Form.

- **Americans with Disabilities Statement & Non-Discrimination Statement:**
  Kean University is an affirmative action, equal opportunity institution. Students with documented disabilities who may need special instructional accommodations or who may need special arrangements in the event of an evacuation should notify the instructor as soon as possible, no later than the second week of the term. Students may contact Kean Disability Office in Downs Hall Rm 122 to discuss special needs, 737-4910.

- **KU Non-Discrimination Policy:** Kean University is an affirmative action, equal opportunity institution.

VIII. Tentative Schedule of Lessons

Note: Some chapter sections will be skipped and announced in class as the semester progresses.

Week	Topics
1	Chapter 1: Linear Equations
2	Chapter 1: Linear Equations
3	Chapter 1: Linear Equations
4	Chapter 1: Linear Equations
5	**Exam 1**, Chapter 2: Matrix Algebra
6	Chapter 2: Matrix Algebra
7	Chapter 2: Matrix Algebra
8	***Spring Recess***
9	Chapter 2: Matrix Algebra, Chapter 3: Determinants,
10	Chapter 3: Determinants
11	**Exam 2,** Chapter 4: Vector spaces
12	Chapter 4: Vector spaces
13	Chapter 4: Vector spaces
14	Chapter 4: Vector spaces, Chapter 5: Eigenvalues and Eigenvectors
15	Chapter 5: Eigenvalues and Eigenvectors
16	No class
17	**Final Exam**

KEAN

Office of Academic Affairs

Course Information

Course Title: Discrete Structures

Course Number and Section: Math 2110, ON16

Semester: *Summer Session I*

Course Meeting Days/Times: Online

Instructor: *Mohib Khan, sakhan@kean.edu*

Course Description: Logic, conditional statements, arguments, predicates and quantifiers, operations and properties of sets, relations defined on sets, equivalence relations, functions defined on sets, counting techniques using multiplication and addition rules, recursively defined relations, graphs and binary trees, regular expressions, automata and applications to computing

Pre-requisite/program: *MATH 1054*

Student Learning Outcomes for the Program (major) aligned with the course objectives:

Program Student Learning Outcomes (SLOs): At the completion of the course student will be able to define operations with set expressions, determine equivalency and non-equivalency of logical statements, discuss properties of relations defined on sets, use counting techniques in solving problems, understand some elementary graph algorithms

Course Objectives: Understand and able to work with discrete mathematics (structures)

Instructional Methods

Method: This course will be taught through Blackboard and Blackboard Collaborate. To access this course go to http://blackboard.kean.edu

- Epp, Discrete Mathematics with Applications, ISBN 978-1337694193

Textbook & Materials

Textbook Title: Epp, Discrete Mathematics with Applications (with WebAssign Access)
ISBN: 978-1337694193
Publisher: Cengage Learning

Open Educational Resources
For more information, please see the Nancy Thompson Library Learning Commons:
http://libguides.kean.edu/OER

Topics and Assignments (Major Topics/themes and readings and assignments that are due each week.)

Unit Number and Title	**Readings/Resources**	**Learning Activities/Point Values**
Unit 1 Logic I	Section 2.1, 2.2, 2.3	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 2 Logic II	Section 3.1, 3.2, 3.4	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 3 Sets	Section 6.1, 6.2	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 4 Functions	Section 7.1, 7.2	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 5 Relations	Section 8.1, 8.2, 8.3	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 6 Counting I	Section 9.1,9.2, 9.3	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 7 Counting II	Section 9.4, 9.5	Discussion-30 Points Assignment-55 Points Quiz-40 Points
Unit 8 Graphs	Section 10.1, 10.2	Discussion-30 Points Assignment-55 Points Quiz-40 Points

Grading Policy

Letter Grade	Out of 1000
A	940-1000
A-	900-939
B+	870 – 899.9
B	840 – 869.9
B-	800 – 839.9
C+	770 – 799.9
C	700 – 769.9
D (Undergraduate Only)	600 – 699.9
F	599 or below

Add (P) Pass or (S) Satisfactory for Undergraduate or (CG) Credit Granted or (NC) No Credit

Important Dates

Mon., May 25	Memorial Day—University closed
Tues., May 26	Term begins
Tues., May 26	Last day to add
Tues., May 26	Last day to drop with 100% refund and with no mark on academic record
Wed., May 27	Mark of "W" begins for course withdrawals
Thurs., May 28	Last day to withdraw with 75% refund
Mon., June 8	Last day to declare a course as an audit
Mon., June 8	Last day to declare or change undergraduate Pass/Fail option
Tues., June 9	Last day to withdraw with 50% refund
Mon., June 29	Last day to withdraw from courses with "W" grade (no refund)
Fri., July 3	Independence Day (Observed)-University Closed
Mon., July 20	Term ends

Discussion Questions Policy

In each unit students will be required to actively participate in the graded Discussions. In these areas of the course a student can interact with the instructor and classmates to explore questions and comments related to the content of this course.

The discussions will be graded for:

1. Frequency: Number and regularity of your discussion comments

2. Quality: Content of your contributions, and

3. Timeliness: When the initial response to the Discussion Question was posted.

Frequency — Number and regularity of contributions. Students are expected to log into the course and post (respond) in each of the unit's discussion topics on a minimum of three separate days that the unit is open. In order to earn full credit six quality posts must be provided during the unit. The student's response to the initial discussion question can count as part of the six required posts.

Quality — Content of the student's contributions. Examples of quality posts include: providing additional information to the discussion, elaborating on previous comments from others, presenting explanations of concepts or methods to help fellow students, presenting reasons for or against a topic in a persuasive fashion, sharing the student's own personal experiences that relate to the topic, and providing a URL and explanation for an area the student researched on the Internet.

Timeliness — Please consult the due dates for details on when responses to DQs are due.

Full credit is awarded only when the quality, frequency and timeliness requirements are met.

Each unit will have Discussion questions. This information comes from the Conversion worksheet. **This text needs to be deleted after this section has been completed.**

Technical Requirements

1. In order for your Blackboard course to function correctly, you will need to disable pop-ups on your Internet browser.
2. Make sure you have Microsoft Office installed on your computer. You may be eligible for a free MS Office Software Student Edition. You are required to create an account and provide a valid Kean University ID to obtain access to the software applications. To start the application process, go to the Office 365 Education website.
3. Download the latest versions of the following:
    - Adobe Acrobat Reader
    - Adobe Flash Player
    - Java JRE

Assessment:

The assessment is based on discussion questions, assignments and quiz in each unit. Each unit consists of 125 points such as DQ 30 points, assignments 55 points and a unit quiz with 40 points. The total points for 8 units are 1000. Please see the grading policy for grade letter out of 1000 points in the above table.

Academic Early Alert information will be provided to students by the instructor via KeanWise. Please see the 2019-2020 Undergraduate or Graduate Catalog for more information.

Important University Policies and Information

All students and faculty should familiarize themselves with the Remote Learning and Virtual Services offered at Kean University via the following link: https://www.kean.edu/coronavirus-information#virus

Students are responsible to review and understand the *University Academic Integrity Policy* available at https://www.kean.edu/academic-integrity

Instructors must indicate on the syllabus and/or individual assignment if the use of Kean University academic support services (Tutoring and/or Writing Center) is permitted for take home assignments/exams.

Students should review the *Student Code of Conduct*, as it discusses expectations of appropriate conduct in the classroom:
https://www.kean.edu/offices/community-standards-and-student-conduct/student-code-conduct

Students are strongly encouraged to register for the University's emergency notification system (http://www.kean.edu/campusalert) in order to be informed of campus emergencies, weather notices, and other announcements. Kean Ocean students would sign up via the following link: https://ocean.sendwordnow.com/LicensePage.aspx

All students must have a valid Kean email account. For those who do not already have one, forms are available on-line at https://www.kean.edu/offices/computer-and-information-services/ocis-forms click on E-mail Account Request Form.

Americans with Disabilities Statement & Non-Discrimination Statement:

Kean University is an affirmative action, equal opportunity institution. Students with documented disabilities who may need special instructional accommodations or who may need special arrangements in the event of an evacuation should notify the instructor as soon as possible, no later than the second week of the term. Students may contact The Office of Disability Services by phone at (908) 291-3137 or email disabilityservices@kean.edu to discuss special needs.

Title IX

Title IX of the Education Amendments of 1972 (Title IX) prohibit discrimination on the basis of sex in education programs or activities. Sexual harassment in any form will not be tolerated at Kean University. Sexual harassment by students should be reported to the Office of Affirmative Action Programs, Office of the Vice President for Student Affairs or the Office of Community Standards and Student Conduct immediately. Information about the University's Sexual Misconduct Policy may be found at the following: https://www.kean.edu/offices/policies/sexual-misconduct-policy

Kean University Non-Discrimination Policy:

Kean University is an affirmative action, equal opportunity institution.

Summer 2020

Office of Academic Affairs

WKU Course Syllabus

Course Information

Course Title:	Fundamentals of Computer Science, Java I
Course Number and Section:	CPS 1231 W04
Semester:	Fall 2019
Course Meeting Days/Times:	W04: Mon. & Wed. 4:00PM - 5:45PM
Course Meeting Location:	C504
Instructor Name:	Prof. Yongfu (Relic) Wang
Office Location:	*C421*
Office Hours:	Mon./Wed. 9:15 - 10:15 AM, 1:30 - 3:30 PM Tues./Thur. 10:00 - 11:00 AM, 2:30 - 4:30 PM (Gladly by Appointment for non-office hours)
WKU Email:	*yongfuw*@wku.edu.cn (or wangyon@kean.edu)

Pre-requisite/program: No Pre-requisite Required.

Course Description: Fundamental computing concepts, components and processes; hardware and software components; communications and information systems; use of systems software; problem solving with application software; introduction to design of algorithms using a high-level programming language.

Course Objectives:
Upon completion of this course, students are expected to:

- Understand key software development methodologies;
- Master the object-oriented programming skill;
- Be familiar with programming using Java IDEs and APIs;
- Be able to solve problems using Java.

Instructional Methods

Methods:
This course is taught using a variety of instructional approaches including lecture, class discussions, small group work, project creation, and electronic discussion (email and Piazza). After-class course discussions are encouraged to take place online on the Piazza:

W04: *piazza.com/kean/fall2019/wkucps1231w04*

Each class session consists of hands-on programming practices at both individual and small group levels. Students are expected to practice while watching instructor's demonstrations during each class sessions.

Take-home programming practices are assigned on a weekly basis. Students are expected to finish them accordingly.

Textbook & Materials:

Textbook Title: Introduction to Java Programming, Brief Version: 11th Edition, By Y. D. Liang
ISBN: 978-1-292-22203-5 (ISBN-13)
Publisher: Pearson

Assessment:

Final numeric score will be weighted as follows:

- Class Participation: 3%
- Quiz: 7%
- Assignments: 20%
- Projects: 20%
- Midterm Exam: 25%
- Final Exam: 25%

-1
-2
-8
-8

Final grade will follow Kean University grading conventions, e.g. A, A-, B+, etc.

Grading Scale:

A	>=94
A-	90-93
B+	87-89
B	84-86
B-	80-83
C+	75-79
C	70-74
D	60-69
F	<60

Academic Early Alert/Mid-Term Progress information will be provided to students by the instructor via KeanWise. Please see the 2019-2020 Undergraduate or Graduate Catalog for more information.

Topics and Assignments

Week	Covered Topics	Textbook Reading	Remarks
1	Introduction to Computers, Program, and Java	Chapter 1	HW1
2	Programming in Java	Chapter 2	HW2
3	Selections	Chapter 3	HW3
4	Functions, Characters, & Strings	Chapter 4	HW4
5	**Quiz 1** Loops	Chapter 5	HW5
6	Loops Loops-Advanced Topic	Chapter 5	HW6
7	Fun Activities Methods Review for Midterm	Chapter 6 Chapter 1 - 5	HW7
8	**Midterm**		Midterm Exam
9	Methods - Advanced Topic Fun Activities	Chapter 6	HW8
10	**Quiz 2** Single - dimensional Arrays	Chapter 7	HW9
11	Single-dimensional Arrays - Advanced Topic Multidimensional Arrays	Chapter 7, 8	HW10
12	Multidimensional Arrays - Advanced Topic JAVA - Advanced Topic	Chapter 8	HW11, Project
13	OOP - Advanced Topic Project Discussion		
14	**Project Presentation** & Final Exam Review		
15	**Project Presentation** & **Final Exam**		

**Topics and Assignments* is tentative. May change accordingly based on overall class performance and progress.

Class Policies

1. Academic Integrity Policy is strictly enforced. https://www.kean.edu/media/academic-integrity-policy

2. Attendance is mandatory for this course. Lack of attendance will be equated with lack of interest in the class and your grade. For legitimate excuses, other than illness, notify instructor at least 7 working days in advance. Students have the responsibility to make up the absent class work.

3. Late homework policy is -20% within 1 late day, -40% within 2 late days and not accepting after 2 late days.

4. It is allowed to discuss assignments with other students. However, if you copied someone
5. else's work, you and the person you copied from will both receive a Zero!

6. You will not pass the course if you failed two or more exams.

7. There is no make-up exam unless the legitimate reason with written evidence.

Important Dates:

See http://www.wku.edu.cn/en/academics/academic-calendar/ for other important dates(e.g. last day to withdraw), or inquire with the Registrar's Office.

Important University Policies and Information:

Students are responsible to review and understand the University Academic Integrity Policy (see http://www.kean.edu/admin/uploads/pdf/AcademicIntegrityPolicy.pdf)

Students should review the Student Code of Conduct, as it discusses expectations of appropriate conduct in the classroom: http://www.wku.edu.cn/en/campus-life/code-of-conduct.

All students must have and use a valid WKU email account. For those who do not already have one, please contact the Registrar's Office.

Disabilities Statement & Non-Discrimination Statement:

Wenzhou-Kean University is an affirmative action, equal opportunity institution. Students with documented disabilities who may need special instructional accommodations or who may need special arrangements in the event of an evacuation should notify the instructor as soon as possible, no later than the second week of the term.

Students may contact Ms. Lin TENG at the Office of the Vice Chancellor for Student Affairs, General Education Bldg., A215, Email: vcsa@wku.edu.cn

Title IX:

Title IX of the Education Amendments of 1972 (Title IX) prohibit discrimination on the basis of sex in education programs or activities. Sexual harassment in any form will not be tolerated at Kean University. Sexual harassment by students should be reported to the Office of Affirmative Action Programs, Office of the Vice President for Student Affairs or the Office of Community Standards and Student Conduct immediately. Information about the University's Sexual Misconduct Policy may be found at the following: https://www.kean.edu/offices/policies/sexual-misconduct-policy.

Student may contact Dr. Shuli XU (shulix@wku.edu.cn), the Title IX contact person at WKU.

KU Non-Discrimination Policy:

Kean University is an affirmative action, equal opportunity institution.

Instructor: Prof. Paolien (Pauline) Wang **Email**: pawang@kean.edu
Class & Lab: Tue, Thur. (12:30-2:25) **Office Hours:** 11:00-12:00

	Date		Topic		Assignment	
1	1/21/2020	Tu	**Take Evaluation Test &** **Evaluation Test Answers discussion:** Elementary programming, Selections, Mathematical functions, Strings + Loops	Ch 1- Ch 5	**Evaluation Test** **Lab #1**	1
	1/23/2020	Th	**Evaluation Test Answers discussion:** Methods, **call by value**	Ch 6	**Program #1**	2
2	1/28/2020	Tu	Single Dimensional Arrays(quick review)	Ch 7	**Lab #2**	3
	1/30/2020	Th	Single Dimensional Arrays, **selection sort/insertion sort**	Ch 8		4
3	2/4/2020	Tu	Multidimensional Arrays	Ch 8	**Lab #3**	5
	2/6/2020	Th	Multidimensional Arrays+ Review for Midterm #1	Ch 8		6
4	2/11/2020	Tu	**Midterm #1 (ch1-ch8)**		**Midterm #1**	7
	2/13/2020	Th	Objects and Classes	Ch 9	**Program #2**	8
5	2/18/2020	Tu	Objects and Classes	Ch 9	**Lab #4**	9
	2/20/2020	Th	Objects and Classes	Ch 9		10
6	2/25/2020	Tu	Thinking in Objects	Ch 10	**Lab #5**	11
	2/27/2020	Th	Thinking in Objects	Ch 10	**Program #3**	12
7	3/3/2020	Tu	Thinking in Objects	Ch 10		13
	3/5/2020	Th	Inheritance & Polymorphism	Ch 11		14
8	3/10/2020	Tu	**Spring break**		**School closed**	
	3/12/2020	Th	**Spring break**		**School closed**	
9	3/17/2020	Tu	Inheritance & Polymorphism	Ch 11	**Program #4**	15
	3/19/2020	Th	Inheritance & Polymorphism	Ch 11	**Lab #6**	16
10	3/24/2020	Tu	Inheritance & Polymorphism	Ch 11		17
	3/26/2020	Th	Abstract Classes & Interfaces	Ch 13		18
11	3/31/2020	Tu	Abstract Classes & Interfaces	Ch 13	**Program #5 due 5/2**	19
	4/2/2020	Th	Recursion, + Review for Midterm #2	Ch 18		20
12	4/7/2020	Tu	**Midterm #2 (ch9-ch11) (ch1-ch8)**		**Midterm #2**	21
	4/9/2020	Th	Recursion	Ch 18	**Lab #7 (recursion)**	22
	4/9/2020	Th	**Last day to withdraw with 'W' grade**		**0% refund**	
13	4/14/2020	Tu	Exception Handling	Ch 12		23
	4/16/2020	Th	Exception Handling & Text I/O	Ch 12	**Lab #8**	24
14	4/21/2020	Tu	Text I/O	Ch 12		25
	4/23/2020	Th	Text I/O, Binary I/O	Ch 17		26
15	4/28/2020	Tu	Binary I/O	Ch 17		27
	4/30/2020	Th	Working on eve (linux system) + Review for Final Exam			28
16	5/5/2020	Tu	**Follow Friday schedule (no Tuesday class)**			
	5/7/2020	Th	**Final exam(ch1-ch13,ch17-ch18)**		**Final Exam**	29
17	5/12/2020	Tu	Review Final Exam Answers			30

For important dates, please consult the Academic Calendar via the following link:
https://www.kean.edu/offices/registrars-office/academic-calendar

Class Information:

- **Course Content:** Fundamental computing and programming concepts; use of systems software; problem solving; design of algorithms using a high-level, object-oriented programming language.
- **A student is required to bring a laptop installed with Eclipse every class.**
- **Prerequisites**: CPS 1231 and Math 1054 (Pre-calculus). Students without prerequisite(s) must withdraw from the class.
- **Instructional Methods**: lecture with slides, class discussions, programming and lab exercises.
- **Textbook**: Introduction to Java Programming (Comprehensive Version), 11th Edition. Y. D. Liang
  Pearson/Prentice-Hall, 2014. ISBN-13: 978-0133761313
- **Objectives**: Upon completion of this course, students will be able to:
  A. Define and describe computing and programming concepts
  B. Use operating system constructs
  C. Demonstrate an understanding of problem solving processes and be able to design algorithms
  D. Design, code, execute, test, debug, and document programs in a high-level, object-oriented programming language
- **Course Grading**: **Midterm1 (25%), Midterm2 (20%), Final Exam (30%), Attendance (3%),**
  8 Lab exercises (10%, Lab1-Lab6: 1%, Lab7-Lab8 1.5%), 4 Programs (8%),
  Term Project (Program 54%), SI sessions (5 extra points)
  A: 100-94, A-: 93-89, B+: 88-84, B: 83-80, B-: 79-76, C+: 75-72, C: 71-68, D: 67-64, below 64 is failing (a C or better is needed for CS and IT majors.)
- **Assignments:** Programming and homework assignments should be coded in Java. You will be able to use the Java complier on the Kean Windows or UNIX-based computers in our classroom. All assigned programmed should be demonstrated on the Kean computers in our classroom.

Class Policies:

- **No late work will be accepted without permission in advance from the instructor. Late work is penalized.**
- **No make-up exams are provided without documentation of medical or family emergency. Please arrange your schedule for this semester to arrive in class on time and prepared.**
- **Attendance, participation and lab exercises are very important part of the final grade. Excused absences must be arranged in advance and evidence of the reason for the absence must be provided.**
- **All work turned in under your name must be your own. No credit will be given for an assignment or homework that is copied – in part or in total – from another person.**
- It is students' responsibilities to verify all the scores posted on **the class website** that is announced on the first day of the class.
- Exam policy: **During class examinations, if a student leaves the classroom for any reason during the exam time, the exam must be turned in before the student departs, concluding the exam for that student. The student cannot resume the exam on return later.** You may not make or take a phone call, text or access the internet during class or a test.
- **Classes may not be audio or digitally captured**, unless permission has been granted by the instructor for that specific class.
- **Public posting of course materials from this class is prohibited**. Materials presented in the classroom are for the personal use of registered students during the semester only. Homework, programs, and other classwork should not be shared, distributed or publicized in any way, for the protection of other students and in respect of the intellectual property rights of the faculty and copyright owners.

Important University Policies and Information:

- The Kean Academic **Calendar** can be found online at https://www.kean.edu/offices/registrars-office/academic-calendar
- **Tutoring and learning Support services**: http://www.kean.edu/content/tutoring-services
- Students should review the **Student Code of Conduct**, as it discusses expectations of appropriate conduct in the classroom: https://www.kean.edu/offices/community-standards-and-student-conduct/student-code-conduct
- Students are responsible to review and understand the University Academic Integrity Policy available via the following link: https://www.kean.edu/academic-integrity
- Read Kean **Students Rights and Responsibilities** at https://www.kean.edu/offices/disability-services/rights-and-responsibilities
- Register for the university's **emergency notification system** at https://www.kean.edu/offices/university-police-0/campus-alert
- Students must have a **valid Kean email account**. You can request one at https://www.kean.edu/~ocisweb/forms.htm
- **Title IX**: Title IX of the Education Amendments of 1972 (Title IX) prohibit discrimination on the basis of sex in education programs or activities. Sexual harassment in any form will not be tolerated at Kean University. Sexual harassment by students should be reported to the Office of Affirmative Action Programs, Office of the Vice President for Student Affairs or the Office of Community Standards and Student Conduct immediately. Information about the University's Sexual Misconduct Policy may be found at the following: https://www.kean.edu/offices/policies/sexual-misconduct-policy
- **Americans with Disabilities Statement & Non-Discrimination Statement:**
  Kean University is an affirmative action, equal opportunity institution. Students with documented disabilities who may need special instructional accommodations or who may need special arrangements in the event of an evacuation should notify the instructor as soon as possible, no later than the second week of the term. Students may contact the Office of Disability Services in Downs Hall Room 122 or call 908-737-4910 to discuss special needs.
- **Kean University Non-Discrimination Policy Statement:**
  Kean University is an affirmative action, equal opportunity institution.

- **Mid-Term Progress** information will be provided to students by the instructor via KeanWise.

Syllabus Updates: **Instructor reserve the right to change the syllabus at any time throughout the semester**. All changes will be announced in class and a new syllabus will be electronically available to all students. If you are enrolled in this class past the add/drop date you are subjected to all rules in this syllabus.

Supplemental Instruction (SI) is available to all CPS 1231 and CPS 2231 students. Regularly scheduled SI sessions involve collaborative learning activities through which students can review notes, clarify material, prepare for tests, and improve learning strategies, all of which will help students master the information and skills of the course. **Regular attendance** at SI sessions has been shown to improve course grades significantly, in addition to building a community of students from your classmates!

SI is a peer-led group study session for CPC 1231 and CPS 2231 courses. SI is led by upper-level students (SI leaders), usually CS and IT majors, who both attend class lectures and hold regularly-scheduled out-of-class sessions (SI sessions). SI sessions provide review or discussion of CPS 1231 and CPS 2231 course concepts, develop skills needed for success on exams, provide an opportunity to meet with classmates, and develop and improve study strategies for course success.

CPS 1231 and CPS 2231 Supplemental Instruction policy:
(a) It is mandatory for students to attend at least 3 SI sessions during the first three weeks of the semester. Regular attendance (1-2 times a week) and participation contributes to academic success. Attendance is taken at SI sessions and reported to CPS 1231/2231 faculty.
(b) Bonus points: Throughout the semester, the SI sessions will provide 5 - 6 challenging questions for students to work on which counts for **5 extra points** reported to your CPS 1231/2231 instructor.

General information, as well as session times and places, will be emailed to students throughout the semester. Suzie Nicolas (nicolsuz@kean.edu, NAAB 214) is also available to assist students in finding their SI sessions.

COMP2521 21T3
Course Outline

Contents

Course Details

Course Code	COMP2521
Course Name	Data Structures and Algorithms
Units of Credit	6
Course Convenor	Dr Ashesh Mahidadia <ashesh@cse.unsw.edu.au>
Lecturers	Dr Ashesh Mahidadia Dr Jiaojiao Jiang
Course Admin	Kevin Luxa
Course Email	cs2521@cse.unsw.edu.au, all course related queries must be sent to this email address.
Course Website	https://webcms3.cse.unsw.edu.au/COMP2521/21T3/
Handbook Entry	https://www.handbook.unsw.edu.au/undergraduate/courses/2021/COMP2521/

Course Summary

The goal of this course is to deepen your understanding of data structures and algorithms and how these can be employed effectively in the design of software systems. It is an important course in covering a range of core data structures and algorithms that will be used in context in later courses. You explore these ideas in lectures, tutorials, lab classes, quizzes and assignments. Assessment involves labs, quizzes, assignments and a final exam involving both practice and theory. At the end of the course, we want you to be a solid programmer, with knowledge of a range of useful data structures and programming techniques, capable of building significant software systems in a team environment, and ready to continue with further specialised studies in computing.

Topics

This course provides an introduction to the structure, analysis and usage of a range of fundamental data types and the core algorithms that operate on them. Key topics are:

- Analysis of algorithms
- Abstract data types
- Binary search trees
- Balanced search trees
- Graphs
- Hashing
- Heaps
- Sorting algorithms
- Text processing algorithms

Executive Summary

A summary of the critical things to know about COMP2521:

- attempt all of the quizzes, labs, tutorials, and assignments yourself
- always try to produce a better program than last time
- in lectures, think critically about what's being said/shown
- the textbook is a useful reference source beyond this course
- assessment:
  - labs: 10%
  - quizzes: 10%
  - assignments: 35%
  - final exam: 45%
- *enjoy the course!*

Now, please read the rest of this document.

Course Timetable

The complete course timetable is available at: webcms3:/timetable

Course Aims

The aim of this course is to get you to **think like a computer scientist**. This certainly sounds like a noble goal... but what does it really mean? How does a *scientist*, let alone a computer scientist, actually think?

What many types of scientists try to do is understand natural systems and processes: a geologist, for example, tries to understand the structure of the earth; a biologist tries to understand living organisms; a chemist tries to understand materials and reactions, and so on.

Computer scientists don't, as the name might suggest, simply try to understand the structure and behaviour of computers, but are more concerned with understanding software systems (and the interaction between the software and the hardware on which it runs). Also, unlike other scientists, computer scientists frequently build the objects that they study.

During this course, we'll be looking at ways of creating, analysing and understanding software. Ultimately, you should be able to answer the question, **"is this piece of software any good?"** and be able to provide sound reasons to justify your answer.

This course follows on from introductory C programming courses: COMP1511, COMP1917, or COMP1921. We cover additional aspects of the C programming language that were not covered in those courses, and also look at some programming tools which were not covered (in detail) earlier. However, this course is not simply a second C programming course: the focus is on the ideas and abstractions behind the data structures and algorithms that are used.

COMP2521 is a critical course in the study of computing at UNSW, since it deals with many concepts that are central to future studies in the area. Whether you are studying Computer Science, Software Engineering, Bioinformatics, Computer Engineering, or even a discipline outside the realm of computing, understanding a range of algorithms and data structures and how to use them will make you a much more effective computing problem solver in the future.

Student Learning Outcomes

After completing this course, students will:

- be familiar with fundamental data structures and algorithms
- be able to analyse the performance characteristics of algorithms
- be able to measure the performance behaviour of programs
- be able to choose/develop an appropriate data structure for a given problem
- be able to choose/develop appropriate algorithms to manipulate this data structure
- be able to reason about the effectiveness of data structures and algorithms for solving a given problem
- be able to package a set of data structures and algorithms as an abstract data type
- be able to develop and maintain software systems in C that contain thousands of lines of code

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

Graduate Capability	Acquired in
scholarship: understanding of their discipline in its interdisciplinary context	lectures, assignments
scholarship: capable of independent and collaborative enquiry	lab work, assignments
scholarship: rigorous in their analysis, critique, and reflection	tutorials
scholarship: able to apply their knowledge and skills to solving problems	tutorials, lab work, assignments
scholarship: ethical practitioners	all course-work, by doing it yourself
scholarship: capable of effective communication	tutorials
scholarship: digitally literate	everywhere in CSE
leadership: enterprising, innovative and creative	assignments
leadership: collaborative team workers	lab work, assignments

Graduate Capability	Acquired in
professionalism: capable of operating within an agreed Code of Practice	all prac work

Assumed Knowledge

The official pre-requisite for this course is either COMP1511 or COMP1917 or COMP1921.

Whether or not you satisfy the pre-requisite, we assume that:

- you can program in the C programming language, and are familiar with arrays, strings, pointers, and dynamic memory allocation
- you are able to design, implement, debug, test and document small C programs (up to several hundred lines of code)
- you are familiar with the Linux environment on the CSE computers

Installing Linux, possibly as a virtual machine, on your own computer would be a major bonus.

Teaching Rationale

Computer science is, to a large extent, a practical discipline, and so COMP2521 has an emphasis on practice. Lectures will include exercises where we examine the practice of developing and analysing programs. The aim of tutorials is to develop analysis and understanding via practical case studies. Lab classes also provide practice in program development and analysis. Assignments provide large case studies of software development.

Teaching Strategies

COMP2521 involves lectures, tutorials, labs, assignments and a text book.

> **Lectures** are delivered live on ***Microsoft Teams Webinar***. The required links for the live lectures will be available on the Lectures, Resources page on Monday of Week-1. Lecture **recordings** will be available later on the Lectures, Resources page.
>
> **Tutorials and labs** are delivered on ***Blackboard Collaborate***. You can access *Blackboard Collaborate* by going to the Moodle page for COMP2521 (click here) and click on the "Tutorials, Labs and Help Sessions" link.

Lectures aim to convey basic information about the course content and to model the practices and techniques involved in software development (*i.e.*, we do demos). The most important components of the course, however, are the tutorials, labs and assignments. Tutorials aim to clarify and refine the knowledge that you got from lectures, and from reading the textbook and notes. Labs and assignments are where you get to put together and practise all of the ideas from the lectures, tutes and text. The only way to develop the skills to do effective software development is by practising them. If you slack off on the assignments and lab exercises (or, worse, rely on someone else to do them for you), you're wasting the course's most valuable learning opportunities.

The university requires us to assess how well you have learned the course content, and the primary approach to achieving this is via a final exam. A final exam is the ultimate summative assessment tool; it gives you a chance, at the end of the course, to demonstrate everything that you've learned. Labs and assignments are a *learning*tool, not an assessment tool, so, in an ideal world, we would have them as pure learning exercises and award no marks for them. However, to give a more concrete incentive to do them (in a timely fashion), there are marks tied to them.

Lectures

Each week, there will be four hours of lectures during which theory, practical demonstrations and case studies will be presented. Lectures convey a small amount of information about the course content, but their main aim is to try to stimulate you to think about concepts and techniques. Lectures will be delivered online via Blackboard Collaborate.

Textbook vs Slides

- **textbook**: contains all material for the course (and more), available from the UNSW Bookstore, describes the material in lots of detail and is very well-written;
- **slides**: the material we use in lectures, available online after the lecture.

Tutorials

Tutorials aim to clarify ideas from lectures and to get you to think about design/analysis issues. There will be a number of exercises set for each tutorial class. The aim of the class is *not* to simply get the tutor to give you the answers; the aim is to focus on just one or two of the exercises and work through them in detail, discussing as many aspects, alternative approaches, fine details, etc. as possible. You must be active and ask questions in tutorials.

Lab Classes

Lab classes aim to give you practice in problem-solving and program development. Each week, there will be one or two small exercises to work on. These exercises will be released in the week preceding the lab class. Labs will be done individually or in pairs, and you and your partner should discuss the exercises *before* going to the lab, to maximise the usefulness of the class. The exercises will need to be submitted (for our records) and will be assessed by your tutor. During the lab, your tutor will provide feedback on your approach to the problem and on the style of your solution.

Important: Although you are required to submit your lab exercises, marks for lab exercises can **only** be obtained by demonstrating your solution to a tutor during your lab session. Simply submitting the lab without demonstrating your solution to a tutor is not worth any marks.

Assignments

In the assignments, you will work on more substantial (hundreds of lines of code) programming exercises. All assignments will be completed individually. As noted above, assignments are the primary vehicle for learning the material in this course. If you don't do them, or simply copy and submit someone else's work, you have wasted a valuable learning opportunity.

Assessment

Your final mark in this course will be based on components from the assignment work, quizzes, labs, and the final exam.

Item	Topics	Due	Marks	Contributes to
Quizzes	All topics	Weeks 2, 3, 4, 5, 7, 8, 9	10%	1, 2, 3, 4, 5, 6, 7
Assignment 1	Trees	Week 7	15%	4, 5, 7, 8
Assignment 2	Graphs	Week 10	20%	4, 5, 7, 8
Labs	All topics	Weeks 1, 2, 3, 4, 5, 7, 8, 9	10%	1, 3, 4, 5
Final Exam	All topics	Exam period	45%	1, 2, 3, 4, 5, 6, 7, 8

Each quiz contributes 2 marks, and we will use your best 6 quiz marks to award marks out of 12, which will be mapped to out of 10 marks. Similarly each lab contributes 5 marks, and we will map your total lab marks to out of 10 marks.

Important: **Labs have no late penalty, because late submissions are not accepted**

The following formula describes precisely how the mark will be computed and how the hurdle will be enforced:

```
quizzes             = mark for quizzes              (out of  10)
labs                = mark for lab exercises        (out of  10)
ass1                = mark for assignment 1         (out of  15)
ass2                = mark for assignment 2         (out of  20)
finalExam           = mark for final exam           (out of  45)

okHurdle            = finalExam > 22.5 (that is, > 50% in the final exam)

mark      = quizzes + labs + ass1 + ass2 + finalExam
grade     = HD|DN|CR|PS  if mark >= 50 && okHurdle
          = FL           if mark <  50
          = UF           if mark >= 50 && !okHurdle
```

Course Schedule

The schedule of lecture topics (subject to change) is:

Week	Topics	Lecturer
1	Analysis of algorithms	Dr Ashesh Mahidadia
2	Recursion, Analysis of ADT (multiple) implementations, Trees	Dr Ashesh Mahidadia
3	Binary search trees (BST), Balanced search trees	Dr Ashesh Mahidadia
4	Search tree algorithms	Dr Ashesh Mahidadia
5	Graph ADT, Graph algorithms (1)	Dr Ashesh Mahidadia
6	...	
7	Graph algorithms (2)	Dr Ashesh Mahidadia

Week	Topics	Lecturer
8	Sorting	Dr Jiaojiao Jiang
9	Hashing, Heaps	Dr Jiaojiao Jiang
10	Tries, Course review and review exercises	Dr Ashesh Mahidadia and Dr Jiaojiao Jiang

Tutorial/Laboratories: Each topic will be dealt with in tutes/labs in the week after it is covered in lectures.

Supplementary Exams

The document **"Essential Advice for CSE Students"** states the supplementary assessment policy for the School of CSE. Please take the time to read it carefully.

If you are granted a supplementary examination, then it will be centrally timetabled. If you think that you may be eligible for a supplementary exam, then make sure you are available on that day. It is your responsibility to check at the student office for details of supplementary examinations.

Resources for Students

COMP2521 follows the contents of the pair of books:

- *Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching* (3rd Edition) by Robert Sedgewick, published by Addison-Wesley
- *Algorithms in C, Part 5: Graph Algorithms* (3rd Edition) by Robert Sedgewick, published by Addison Wesley

These two books are available as a bundle from the UNSW bookshop. They are expensive, but are useful well beyond this course, and will serve as a useful reference on the bookshelf of any serious programmer.

You may also be able to find on-line resources related to the textbooks. Robert Sedgewick has a series of videos on the topics in this course, but unfortunately they all seem to be in Java (which he has used for the new edition of his book). If you find any useful on-line resources, please let me know and we will add them to the *Resources* section of the course web site (with credit to the finder).

This website also has links to the auxiliary material/documentation that you will need for the course. Solutions for all tutorial questions and lab exercises will also be made available. We will review quiz and assignment solutions in the lectures.

Student Conduct

The **Student Code of Conduct** (Information, Policy) 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 of Conduct 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, the School Grievance Officer, or one of the student representatives.

Plagiarism is defined as 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
- UNSW Plagiarism Procedure

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.

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

- Student Code Policy
- Student Misconduct Procedure
- Plagiarism Policy Statement
- Plagiarism Procedure

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

- Essential Advice for CSE Students

Course Evaluation and Development

Student feedback on this course, and on the effectiveness of lectures, tutorials and labs in this course, is obtained via electronic survey (myExperience) at the end of each semester. Student feedback is taken seriously, and continual improvements are made to the course based in part on this feedback. Students are strongly encouraged to let the lecturer in charge know of any problems

as soon as they arise. Suggestions and criticisms will be listened to openly, and every action will be taken to correct any issue or improve the students' learning experience.

This term, our focuses will include:

- improve learning experience during tutorials and labs by encouraging student participation.
- encourage active learning during live lectures.

COMP2521 21T3: Data Structures and Algorithms is brought to you by
the School of Computer Science and Engineering at the University of New South Wales, Sydney.
For all enquiries, please email the class account at cs2521@cse.unsw.edu.au
CRICOS Provider 00098G

ALGORITHMS
COMP3121 | COMP9101
COURSE INFORMATION

Home
Course Information
Staff Contact Details
Course Textbook
Assessment
Course Schedule
Course Syllabus
Assignments, Exams and Puzzles
Course Resources
Course Evaluation and Development
Academic Honesty and Plagiarism

UNSW AUSTRALIA | University of New South Wales
School of Computer Science and Engir

Course Syllabus

Topics to be covered in COMP 3121/9101 & COMP 3821/9801 include*:

Introduction:

- Please review the basic material from Aleks's review manual

Algorithm Analysis:

- Proving correctness of algorithms
- Stable Matching Problem

Divide-And-Conquer Method:

- Asymptotic behavior, recurrences, summations, estimations;
- Weighing coins
- The Master Theorem;
- Applications of the master Theorem: Fast integer multiplication;
- When should you trust asymptotic estimates?
- Multiplying polynomials and FFT

The Greedy Method:

- When greed pays off - foundations of the Greedy Method;
- Activity Selection problem;
- Discrete (0-1) Knapsack Problem;
- File compression: Huffman Codes.
- Directed acyclic graphs and topological sorting
- Dijkstra's algorithm
- Minimum spanning trees

Dynamic Programming Method:

- Longest Increasing Subsequence
- Making change
- Assembly line scheduling;
- Multiplying chains of matrices;
- Longest Common Subsequences;
- Edit distance
- Bellman – Ford algorithm.
- Floyd – Warshall algorithm.

Network Flow Algorithms:

- Flow networks;
- Ford – Fulkerson method;
- Applications including Maximum Bipartite Matching

String Matching Algorithms:

- "Naïve" string matching algorithm;
- Rabin – Karp algorithm;
- String matching using finite automata

Linear Programming:

- Formulating linear programs
- Linear programming and integer linear programming

Intractable Problems and Approximation Algorithms:

ALGORITHMS
COMP3121 | COMP9101
COURSE INFORMATION

Home
Course Information
Staff Contact Details
Course Textbook
Assessment
Course Schedule
Course Syllabus
Assignments, Exams and Puzzles
Course Resources
Course Evaluation and Development
Academic Honesty and Plagiarism

- Feasibility of algorithms
- Polynomial Time problems, NP problems, intractable problems
- NP complete problems and NP hard problems
- Feasible reductions
- Approximate solutions using Greedy Method and DP

Practicing problem-solving while having fun:

- Assorted puzzles

Privacy statement Copyright and Disclaimer Site Map Site Feedback
© Algorithm Course, CSE, UNSW, Sydney NSW 2052, Australia. | Design by Mohsen Rezvani

Resources / Course Outline

Course Outline

Contents

Course Details

Course Code	COMP1511
Course Title	Programming Fundamentals
Lecturer/Convenor	Sasha Vassar (/users/z3059042)
Admin	Tom Kunc (/users/z5205060) and Shrey Somaiya (/users/z5257343)
Lectures	**Tuesday** 10am-12pm online **Wednesday** 2pm-4pm online Details for online lectures are below
Tutorials and Labs	Timetable for all classes (https://webcms3.cse.unsw.edu.au/COMP1511/21T3/timetable)
Help Sessions	Online Help Sessions will be timetabled soon . . .
Course Contact Email	cs1511@cse.unsw.edu.au
Units of Credit	6
Course Website	http://cgi.cse.unsw.edu.au/~cs1511/21T3/ (http://cgi.cse.unsw.edu.au/~cs1511/21T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1511.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1511.html)

Course Summary

This course is an introductory course into the basics of Computer Programming and Computer Science. It is intended as an introduction to studying further in Computer Science or related fields. Topics include:

- Fundamental programming concepts

- Introduction to Computer Science
- The C programming language and use of a C compiler
- Programming style
- Program design and organisation concepts
- Program testing and debugging

Assumed Knowledge

Before commencing this course, students should:

- Have basic computer literacy (not necessarily have programmed before)

Otherwise, COMP1511 assumes no background knowledge.

Student Learning Outcomes

This course aims for students to become proficient in a high-level programming language, C. It also focuses on mental preparedness for programming long term, including problem solving, debugging and testing.

After completing this course, students will:

- Have basic proficiency with the C programming language
- Have the ability to analyse a problem and solve it using programming
- Have learnt some techniques for debugging and testing code and programs
- Understand how to use basic data structures like arrays and linked lists
- be able to use the basics of a Linux-like, command line driven operating system

Teaching Strategies

This course has a heavy practical orientation. Lectures will revolve around live demonstrations of programming and use of tools. Labs and assignments are also highly practical.

On top of this, the course is not just about the specific technical aspects of Programming, but also a preparation for studying Computer Science and the thought processes and skills necessary for a career in the field.

Lectures

Lectures will be used to present the theory and practice of the techniques and tools in this course. There will be extensive use of practical demonstrations during lectures. Lecture notes will be available on the course web pages before each lecture.

Lectures will be delivered as YouTube Live streams.

All lectures will be recorded. For anyone who cannot access the live stream or who cannot access them live, lecture recordings will be made available.

There are no course related lectures in Week 6 (Flexibility Week), however, we often run guest lectures covering different topics in programming that are optional and may be of interest

Live Streaming and Videos
In addition to scheduled lectures, there may be some more informal streams and pre-recorded videos to give students a chance to ask questions directly, as well as, cover content that is not "official" course content, but still might be useful.

Tutorial/Lab Sessions

From week 1 you will also be expected to attend a three hour tutorial/laboratory session to clarify ideas from lectures and work through lab exercises, based on the lecture material. You should make sure that you use this time effectively by ***examining in advance*** the material to be covered in each week's tutorial. This means that you are coming to class prepared to ask any questions that you may have, and generally participate in class by offering suggestions - this will ensure that you get the most possible out of the tutorial/lab session. Your tutors are there to help you clear up any misunderstandings or to understand topics in more depth. The tutorial questions will be linked to the class webpage in the week before each tutorial. There are no marks for tutorial attendance, however, it is your chance to have all your questions answered.

Due to the current COVID-19 restrictions, tutorial/labs will all be run online via a system called Blackboard Collaborate. Links for Tutorial/Lab sessions will be available on Moodle. We will notify you if there are opportunities for face-to-face classes to resume, based on the government and university restrictions at any given time.

Following the theoretical section of the Tutorial/Lab session, there will be time to work on practical exercises, as well as, have some time to have one-on-one conversations with your tutors to get specific help.

Because this course is practical in nature, laboratory classes are a very important component. If you do not put a great deal of effort into the lab classes you risk failing the final exam. Please use laboratory sessions to ask tutors for any help you may need if you are stuck on that week's problems.

There are also some Challenge Exercises for anyone looking to push beyond the course content, or see some interesting continuation of the content in the course. They are not necessary to complete.

Lab Submission and Marking
Each lab exercise will be submitted using the "give" system. All students will need to submit solutions.

If you cannot complete any exercises by the end of the lab, you may complete them in your own time and submit them using the "give" command before 8pm Monday (Monday 20:00 AEST) in the week after the lab.

Labs will be marked automatically around a week after the due date. When marking is complete you can see marks online here (http://cgi.cse.unsw.edu.au/~cs1511/21T3/student/) .

There are no marks for Lab 1, it's there to help you get started. Your total mark will be made up of the best 7 out of 8 of the labs from weeks 2-10 (there is no lab in week 6).

Lab exercises have an indicative level of complexity:

Indicator	Level of Complexity	Marks total per week
●○○	Every student in the course should be able to complete these. Students who don't complete these may struggle to pass the course. If you are having trouble completing these exercises, please contact us or come along to the help session, so that we can help you.	1 mark per week
●●○	Every student in the course should attempt these. These questions may take slightly longer to solve, but will help to link together concepts taught in lectures and tutorials with the practical components. Being able to solve these questions will be helpful in assignments, the exam, or in future courses.	0.5 marks per week
●●●	These exercises are for students who want to be exposed to more complex applications of content they've been taught. While it's not essential, students who complete these exercises are better prepared for later parts of assignments and exams. These exercises usually require a lot more time to solve and complete.	0.5 marks per week

☠	These exercises are for students who want to extend themselves. They may not necessarily teach you more programming, but will expose you to difficult problems and interesting parts of Computer Science, where you will get a good opportunity to practice some killer problem solving skills!	no marks, only for challenge.

Help Sessions

There will be consultation sessions starting around week 3 or 4,where tutors will be available for one-on-one help with specific problems and assignment clarification. These sessions are optional and will run at different times during the week, with more sessions available around assignment deadlines and in later weeks of the term. Check the course timetable for what Help Sessions have been scheduled.

Weekly Coding Tests

There will be 7 weekly coding tests from weeks 3-5 and 7-10 designed to give you timely & realistic feedback on your understanding of the course material.

When you commence the weekly test, please time yourself for one hour (self-enforced exam conditions). This gives you both accurate feedback on your progress as well as some practice for coding under time constraints. You will still have there rest of the week to work towards solving the weekly test exercises.

During your self-enforced conditions, typically:

- No assistance from any person, or asking questions online.
- Try to set a time limit (1 hour), so you can see where your progress is at.

This will give you a reasonable idea of how you're going in the course at this time and what topics you might want to study further.

After you finish the time limit and the self-enforced exam conditions, you then have a lot more time to complete the exercises if that first hour was not enough. You can then continue working on the questions with whatever resources you'd like to use. The only difference is: We won't be discussing the weekly tests on the forums until after everyone's had a chance to complete them. We don't want to spoil other people's feedback by giving them the answers too early.

Each coding test will be automatically marked. Partial marks will be awarded based on the percentage of marking tests passed. These tests will be similar to the automated tests provided to you but may differ slightly.

Marks for the coding test component will be the sum of the best 6 of 7 test marks.

Any deliberate violation of the test conditions will result in a mark of zero for the entire programming test component.

The weekly programming test must be completed by Thursday 8pm the week after it is released.

Assignments

There are two assessable programming assignments. Assignments give you the chance to practice what you have learned on relatively large problems (compared to the small exercises in the labs). Assignments are a very important part of this course, therefore it is essential that you attempt them yourself. Collaboration with other students is limited to discussion of fundamentals, not any discussion of assignment specifics.

- Assignment 1 (Submission in Week 6) 15%
- Assignment 2 (Submission in Week 10) 25%

The assignment weighting and deadlines may vary slightly when the assignment designs are complete.

Late assignments submissions will be penalized. The exact penalty will be specified in the assignment specification - typically it is 2% reduction in maximum possible mark for every hour late.

Final Exam

In 2021 Term 3, the Exam will be a 6 hour long take-home exam. The actual working time of the exam is only expected to be around 3 hours, but due to time zone differences for current students, all students will be given a longer time window to complete the exam.

It will contain implementation tasks that will require you to write C programs. It will also contain sections which require you to read code or answer questions to show your knowledge of programming.

During this exam you will be able to execute, debug and test your answers. The implementation tasks will be similar to those encountered in lab exercises and Weekly Tests.

Special Exam Requirements
COMP1511 has two requirements on the final exam.

Requirement#1: on the final exam you must solve a task by writing a program that uses an array. There will be multiple, clearly marked, questions that will involve the use of an array. You must answer one of these questions to meet this requirement.

Requirement#2: on the final exam you must solve a task by writing a program that uses a linked list. There will be multiple, clearly marked, questions that will involve the use of a linked list. You must answer one of these questions to meet this requirement.

You can not pass COMP1511 unless you achieve both the above requirements.

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 enrollment 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 online 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	Topics	Due	Marks Released	Marks
Lab Exercises	All topics	Weeks 2-10	1 week after they are due.	10%
Weekly Tests	All topics	Weeks 3-10	1 week after they are due.	10%
Assignment 1	Looping and Arrays	Week 6	Between weeks 8 and 9.	15%
Assignment 2	Linked Lists	Week 10	**After** the final exam is sat.	25%
Final Exam	All topics	Exam period	Before the T3 holidays.	40%

Course Schedule

Week	Lectures	Tutes	Labs	Live Streams	Assignments	Weekly Test
1	Course intro, Our First C Program (variables and if statements)	Welcome, What is programming?	Lab familiarization, Setting up working from home (VLAB), Basic Input/Output	-	-	-
2	Structs, Looping and Code Style	Variables and If Statements	Variables and If Statements	-	-	-

3	Functions and Arrays	Structs and Looping	Structs and Looping	-	-	Basic Input/Output, if statements
4	Functions, Arrays, Memory and Pointers	Functions and Arrays	Functions and Arrays	Assignment 1 overview	Assignment 1 released	Looping
5	Debugging, Strings, Character functions and Multi-file projects	Functions, Arrays and Pointers	Functions, Arrays and Pointers	-	-	Functions and Arrays
6	Flexibility Week. Guest Lectures, Halfway Course recap.	No Tutorials	No Labs	-	Assignment 1 due	No Test
7	Memory Allocation, and Linked Lists	Characters, Strings	Characters, Strings	-	-	Functions, Arrays and Pointers
8	Linked Lists	Memory Allocation and Linked Lists	Memory Allocation and Linked Lists	Assignment 2 overview	Assignment 2 released	Characters, Strings and structs
9	Abstract Data Types, Recursion	Linked Lists	Linked Lists	-	-	Memory and Linked Lists
10	Exam prep and Course recap	Abstract Data Types	Exam practice (past exam questions)	-	Assignment 2 due	Past exam questions
11	Revision and study for the exam			Revision stream	-	-

Resources for Students

There is no requirement for a text book for COMP1511. Generally, students do not purchase this textbook. It covers material in a different way to the COMP1511 course materials, and goes into differing levels of content than this course. It may, however, be useful as a reference, or to explore some content in more detail.

The optional textbook for the course is: *Programming, Problem Solving, and Abstraction with C* by Alistair Moffat (http://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9781486010974) , ISBN 978 1 74103 080 3, which can be purchased from the UNSW Bookshop.

Course Evaluation and Development

At the end of every term, COMP1511 students are invited to provide their feedback about the course through the UNSW myExperience online survey system. This is used to assess the quality of the course in order to make on-going improvements. We do take this feedback seriously and use it to improve the course materials and their delivery. Students are also encouraged to provide informal feedback during the session, and to let the lecturer in charge or any of the course staff, know of any problems, as soon as they arise. Suggestions will be listened to very openly, positively, constructively and thankfully, and every reasonable effort will be made to address them. Recent MyExperience evaluations showed that students were highly satisfied with most aspects of the course. However, there are always things that can be improved, some changes that we are making this term:

- In particular, students really enjoyed tutorials/labs and found they learned a lot. We still have some work to do on more interactive teaching in labs, however, we have taken on a lot of feedback in 21T2 and will be trialling supplementary short videos to help with lab and tutorial questions.

- Additionally, students found the assignment specifications to be long and sometimes overwhelming, these will be overhauled this term to create a more streamlined approach and to help students with understanding the problem. Specifications will be broken down into more understandable stages and students will get an understanding of stage difficulty in the assignment.

- As per student feedback, to relieve some of the pressure on the high complexity material covered in the latter part of the course, we are trying to spread out the complexity across the first half of the course - shifting around some topics, such as Structures and chars to the first half of the term.

- Student feedback further indicated that the variable number of lab exercises was hard to approach, especially in weeks, where other things were also due. To combat this, we have streamlined the number of exercises across the different working weeks. Exercises also now give an indication of difficulty and marks associated with them for each of the weeks.

CSE may also run its own survey, midway through the term, to elicit feedback while courses are still running. This course improves only because we see the difficulties that students have and try to adjust things so that you get to learn what you need. If anything's not working for you, please let us know and we'll do whatever we can to help and hopefully help students in later cohorts as well.

Resource created 5 months ago (Tuesday 24 August 2021, 01:37:36 PM), last modified 4 months ago (Tuesday 28 September 2021, 04:38:55 PM).

Comments

(/COMP1511/21T3/forums/search?forum_choice=resource/64999) (/COMP1511/21T3/forums/resource/64999)

There are no comments yet.

Course Outline

Contents

Course Details

Course Code	**COMP3411/9814**
Course Title	**Artificial Intelligence**
Units of Credit	**6**
Course Website	**http://cse.unsw.edu.au/~cs3411 (http://cse.unsw.edu.au/~cs3411)**
Handbook Entry	**http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3411.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3411.html)**
Lecturer-in-Charge	Claude Sammut <cs3411@cse.unsw.edu.au>
Course Admin	Armin Chitizadeh <cs3411@cse.unsw.edu.au>

Timetable

	Monday	Tuesday	Wednesday	
09:00 - 10:00				

10:00 - 11:00				Tut - H10A Adam Stucci
11:00 - 12:00				Tut - H11A (online) (https://moodle.telt.unsw.edu.au/mod/lti id=4482603) Jingying Gao
12:00 - 1300				Tut - H12A (online) (https://moodle.telt.unsw.edu.au/mod/lti id=4482603) Lina Phaijit
13:00 - 14:00				Tut - H13A (online) (https://moodle.telt.unsw.edu.au/mod/lti id=4482603) Armin Chitizadeh
14:00 - 15:00				Tut - H14A Raktim Kumar Mondol
15:00 - 16:00				Tut - H15A Jingchen Li
16:00 - 17:00			Lecture (initially online but may later switch to synchronised in-person and streaming, depending on progress of pandemic)	Tut - H16A Jingchen Li
17:00 - 18:00				Tut - H17A Maryam Hashem
18:00 - 19:00			Tut - W18A (online) (https://moodle.telt.unsw.edu.au/mod/lti/view.php?id=4482603) Ryan De Bellen	Tut - H18A Maryam Hashem

Course Summary

Artificial Intelligence is concerned with the design and construction of computer systems that "think". This course will introduce students to the main ideas and approaches in AI - including agent architectures, AI programming, search techniques, knowledge representation and reasoning, machine learning, natural language processing, logical inference and robotics.

This course will introduce you to the following main ideas and approaches in AI, which will be presented in four main modules:

1. *Artificial Intelligence, Intelligent Agents*

- Introduction and history of AI
- Agents and Autonomous Systems

2. *Knowledge and Reasoning*

- Path Search
- Heuristic Path Search
- Constraint Satisfaction
- Logical Agents and Logic Programming
- Uncertainty

3. Machine *Learning*

- Learning and Decision Trees
- Reinforcement Learning and Neural Networks

4. *Additional topics:*

- Natural Language Processing
- Computer Visions
- Robotics

Assumed Knowledge

COMP3411

Pre-requisite: COMP1927 or COMP2521

COMP9814

Pre-requisite: COMP9024

Student Learning Outcomes

Students successfully completing this course will:

- Gain a working knowledge of fundamental AI methods and techniques.
- Be able to demonstrate knowledge of AI methods by explaining certain features or aspects of these algorithms.
- Describe how the techniques would be applied to particular problems.
- Develop competency in the AI programming.
- Gain practical experience through the assignments to understand what is involved in designing and implementing a functional AI system.

Teaching

There will be four hours of lectures per week, plus one hour of tutorial. The major AI algorithms and learning techniques will be presented in lectures and illustrated on sample problems, along with historical background and theoretical motivation.
Tutorials give students a chance to clarify the ideas mentioned in lectures and practice their problem-solving skills in a small (and hopefully more personal) class with the assistance of a tutor. Students are expected to prepare for and actively participate in tutorials. Most tutorials will also include one or two questions of a

speculative nature - which can lead to more in-depth discussion of particular topics, depending on the interests of the students.

Assessment

The assessable components of the course are:

Component	Mark
Assignments	40%
Written Exam	60%

Further details about the assignments will be posted on the Course Web site. Programming assignments give the students an opportunity to put into practice the ideas and approaches that have been presented in lectures and discussed in tutorials. They may, for example, involve writing a program to:

- enable an agent to act in a simulated environment
- solve a problem using search techniques
- play a game
- apply a machine learning algorithm
- enable communication or co-operation between agents

In order to pass the course, students must score:

- at least 16/40 for the assignments
- at least 24/60 for the exam
- a combined mark of at least 50/100

Resources for Students

The recommended textbook for this course is:

- David L. Poole and Alan K. Mackworth *Artificial Intelligence: Foundations of Computational Agents* , 2nd Edition (https://artint.info/2e/html/ArtInt2e.html)

There is an electronic version of the book, as well as print. Here are the links to both:

- Print - https://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9781107195394 (https://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9781107195394)
- (https://www.bookshop.unsw.edu.au/details.cgi?ITEMNO=9781107195394) Digital - https://artint.info/2e/html/ArtInt2e.html (https://artint.info/2e/html/ArtInt2e.html)

The following books might also serve as additional reference material:

- Stuart Russell and Peter Norvig, *Artificial Intelligence: a Modern Approach* , 3th Ed., Prentice Hall, 2010.
- Ivan Bratko, *Prolog Programming for Artificial Intelligence* , 4th Edition, Pearson, 2013.
- Nils J. Nilsson, *Artificial Intelligence: a New Synthesis* , Morgan Kaufmann, 1998, ISBN 1-55860-467-7.
- Valentino Braitenberg, *Vehicles: Experiments in Synthetic Psychology* , MIT Press, 1984, ISBN 0-262-52112-1.

Links to other electronic resources will be provided on the Course Web page throughout the session.

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)

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

Resource created 2 months ago (Thursday 03 February 2022, 04:11:57 PM), last modified about a month ago (Wednesday 16 February 2022, 04:47:05 AM).

Comments

(/COMP3411/22T1/forums/search?forum_choice=resource/71579)

(/COMP3411/22T1/forums/resource/71579)

Add a comment

There are no comments yet.

Course Outline

Contents

Course Details

Course Code	COMP3311
Course Title	Database Systems Implementation
Convenor	Dong Wen (/users/z3527094)
Admin	Qingshuai Feng (/users/z5245537)
Classes	**There are no on-campus lectures in COMP3311 in 22T1.** We have several tutorials will be held offline. Timetable for all classes (/COMP3311/22T1/timetable)
Consultations	TBA
Units of Credit	6
Course Website	https://webcms3.cse.unsw.edu.au/COMP3311/22T1/ (https://webcms3.cse.unsw.edu.au/COMP3311/22T1/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3311.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3311.html)
Issues	email cs3311@cse.unsw.edu.au (mailto:cs3311@cse.unsw.edu.au) (mailto:dong.wen@unsw.edu.au)

You are invited to meet us during consultation time slots and during the tutorial periods if you have any question. You are also welcome to use the forum, contact us via the course email, or contact lecturer (**dong.wen@unsw.edu.au**) for urgent queries.

Course Summary

This course aims to explore in depth the practice of developing database applications and the theory behind relational database systems. It will also give a very brief overview of the technologies used in implementing database management systems and the past, present and future of database systems.

Large data resources are critical to the functioning of just about every significant modern computer application. Hence, knowledge of how to manage them is clearly important to the IT industry. In the context of further study, understanding how to use databases effectively is essential for courses such as COMP9321 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9321.html) Data Services Engineering and COMP9322 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9322.html) Software Service Design and Engineering. COMP3311 also provides a foundation for further study in advanced database topics, such as COMP9312 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9312.html) Graph Data Analytics, COMP9315 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9315.html) Database Systems Implementation and COMP9318 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9318.html) Data Warehousing and Data Mining. Database concepts are also relevant in courses such as COMP9319 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9319.html) Web Data Compression and Search and COMP6714 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6714.html) Information Retrieval and Web Search.

By the end of this course, we want you to be capable of building high-quality (correct and efficient) applications based on relational databases, to have a sound understanding of issues in managing relational database management systems, and an overview of how they work internally.

Assumed Knowledge

The official pre-requisite for this course is that students have taken either COMP2521 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2521.html) or COMP1927 (http://legacy.handbook.unsw.edu.au/undergraduate/courses/2018/COMP1927.html) .

Whatever the formal pre-reqs, we assume primarily that students have some experience with procedural programming and some knowledge of data structures.

A perpetual problem for COMP3311 is that around half of the class has already covered basic data modelling techniques (specifically ER diagrams) and/or basic SQL in courses such as INFS1603 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/INFS1603.html) Introduction to Business Databases or COMP1531 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1531.html) Software Engineering Fundamentals. On the other hand, half of the class hasn't seen this material before, so we need to cover it. Those who have seen it before should treat this as revision. Don't make the mistake of thinking "I know all this stuff"; we will definitely cover these areas in more depth than you have seen them previously.

Student Learning Outcomes

On successfully completing this course, students should be able to:

1. develop accurate, non-redundant data models;
2. realise data models as relational database schemas;
3. formulate queries via the full range of SQL constructs;
4. use stored procedures and triggers to extend DBMS capabilities;
5. write applications in Python that interact effectively with databases;

6. analyze performance issues in relational database applications;
7. understand the overall architecture of relational DBMSs;
8. understand the concepts behind transactions and concurrency control;

Glossary:

- **DBMS** : DataBase Management System ... software system to support database manipulation.
- **RDBMS** : Relational DBMS ... the most popular style of DBMS (refers to underlying data model) .
- **SQL** : Structured Query Language ... the ANSI standard language for manipulating RDBMSs.

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

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	assignments, prac work
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	-
Professionals capable of ethical, self- directed practice and independent lifelong learning	assignments, prac work
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	-

Teaching/Assessment Strategies

Summary of teaching strategies in COMP3311 22T1.

- Lectures ... introduce concepts and work through examples
- Tutorials ... reinforce concepts and provide additional examples
- Prac Work ... introduce technology required for the assignments
- Assignments .. allow students to solve significant problems
- Quizzes ... review concepts from previous week
- Forums ... ask questions, get answers
- Consultations ... ask questions (live), get answers

Online Lectures

In timetabled lecture slots, we will introduce several topics and run some online problem-solving sessions where we will work together through problems related to the content for that week. All lectures will be recorded.

Tutorials

Each week, you will have a 90-minute online session (via Moodle Blackboard Collaborate) with a tutor. In this term, we also have several offline tutes allowing you to have a F2F communication with your tutors. There will be a set of tutorial problems available each week, covering topics from the previous week. During the tute, you can either run a Q&A with the tutor or work through some of the problems together. Discussion is encouraged. Solutions will be made available to the tutorial problems in the weeks following the tute.

Prac Exercises

Prac exercises will be made available periodically on the course web site. You should work through each exercise when it becomes available, since exercises are often a pre-cursor to the assignment work that follows.

Assignments

In the assignments, you will consider the major stages of the database application development process, conceptual design, implementation as a PostgreSQL application, and building components of a database application using Python. The assignments contribute 35% of the overall mark for this course.

Assignments are completed *individually* ; this means that you should do them *yourself* without assistance from others, except by asking advice from the Lecturer or Course Admin or your Tutor. Assignments are a primary vehicle for learning the material in this course. If you don't do them, or simply copy and submit someone else's work, you have wasted a valuable learning opportunity. The assignments also lay the foundation for successful performance in the final exam.

Assignments are to be submitted on-line (via Moodle) before midnight on the due date. Assessment of assignments will be based on how accurately they satisfy the requirements, as determined by automatic marking, and by whether they meet efficiency requirements. We provide testing harnesses for the assignments so that you can determine whether your code is producing the correct output. The supplied tests will use one instance of the database; the auto-marking will use these same tests, but will also run tests using one or more different database instances.

Late submissions will have marks deducted from the maximum achievable mark at the rate of 0.1 of the total mark per hour that they are late.

Quizzes

Starting in Week 2, there will be quizzes on topics from previous weeks. This gives you a chance to review what you've learned on those topics. Quizzes are released on Monday mornings and are due before midnight on the following Friday (i.e. 5 days later). Each quiz will have 4 questions.

Do *not* submit your quiz answers late (there are heavy late penalties), or forget to submit a quiz.

Forums/Consultations

You should make use of the forums to help you with prac work and assignments. If you want your question answered, you need to provide details (e.g. what you tried, what you observed, the context where you tried it). It is difficult to provide assistance without enough information on what you're doing. In particular, problems in setting up and running your PostgreSQL server are very difficult to diagnose with a lot of context. Such questions are probably best handled in a consultation, where you can share your screen.

One thing you should *not* post on the forum are examples of assignment work (e.g. "Tried this query for Q3, but it gave me this unexpected output"). Save such questions for consultations.

Final Exam

The Final Exam will be held online. The paper will be made available via Webcms3 and submission will be via Moodle. The paper itself will be primarily based on SQL and PLpgSQL. It should take approximately 3-hours to complete. The exam will be open for a 4-hour time-period.

Note: anyone caught colluding, collaborating, posting exam solutions onto any internet platform, reading solutions from any internet platform, will be reported to the Student Conduct Investigation Unit, and will most likely end up with a mark of 00FL for the course, if not exclusion from further study at UNSW.

Teaching Rationale

COVID has thrown up some challenges in running CSE courses. We have adopted the above strategies because we believe they are the best way for helping you to learn in the online environment.

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. This encompasses both copying work from your fellow students (plagiarism) or asking/paying someone to do the work for you (contract cheating).

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)

Assessment

There will be three places where your learning in this course will be assessed: **assignments** , **quizzes** and the **final exam** .

Much as we dislike conflating the *learning* aspect of assignments with their *assessment* aspect, there will be marks for the assignment work. We would rather use the assignments entirely as learning vehicles and have no assessment associated with them, but I suspect that wouldn't result in a satisfactory outcome (i.e., nobody would do them).

There will also be a number of on-line quizzes during the semester, to be taken in your own time (but before indicated due dates), primarily as a way for you to gauge your progress in the course, but also worth some marks.

Lastly, the final exam will form the major assessment in this course and aims to test what you learned about databases during the course of the semester. The Final Exam will be held online, and will involve both a practical component (writing SQL queries on a supplied database) and a theory component (e.g., data modelling, schema normalization, etc.). In order to meet the exam hurdle requirement, you must score at least 40% (20/50) on the exam. Note that partial marks are available for all questions on the final exam, and the hurdle will be enforced after any required scaling.

The exam will contain questions that your should be able to comfortably answer in 3 hours if you know the material well. However, you will have a 4-hour window in which to complete the exam.

Item	Topics	Due	Marks	Contributes to
Quizzes	All topics	Weeks 2,3,4,7,8,10	15%	1,2,3,4,5,6,7,8
Assignment 1	SQL/PLpgSQL	Week 5	15%	3,4
Assignment 2	Python/SQL	Week 9	20%	5
Final Exam	All topics	Exam period	50%	1,2,3,4,5,6,7,8

Your final mark in this course will be based on the marks from the above three assessment components. Note that the exam is a hurdle, so that if you fail the exam badly enough, you cannot pass the course. The following formula describes precisely how the final mark will be computed and how the hurdle will be enforced:

```
quizzes = mark for on-line quizzes   (out of 15)
ass1    = mark for assignment 1      (out of 15)
ass2    = mark for assignment 2      (out of 20)
exam    = mark for final exam        (out of 50)
okExam  = exam >= 20                 (after scaling)
mark    = ass1 + ass2 + quizzes + exam
grade   = HD|DN|CR|PS  if mark >= 50 && okExam
        = FL           if mark <  50 && okExam
        = UF           if !okExam
```

Course Schedule

Week	Lectures	Tutes	Prac Work	Assignments	Quizzes	Notes
1	Course intro, Data Modelling, ER diagrams	-	SQLite3, Set up PostgreSQL server	-	-	-
2	Relational Model, SQL DDL, Mapping ER to SQL	Data modelling, ER	Defining a database	-	Quiz 1	-

3	SQL queries	ER -> SQL	SQL queries	Ass 1 released (mon)	Quiz 2	-
4	PLpgSQL functions	SQL	PLpgSQL functions	-	Quiz 3	-
5	Triggers, Aggregates	PLpgSQL functions	Aggregates	Ass 1 due (fri)	-	-
6	Flexibility Week	-	-	-	-	-
7	DB/PL interaction, Python, Psycopg2	Triggers	-	Ass 2 released (mon)	Quiz 4	-
8	Functional dependencies, Normalization	Psycopg2	-	-	Quiz 5	-
9	Relational Algebra, Query execution, Tuning	Normalization	-	Ass 2 due (fri)	-	-
10	Transactions, Concurrency control, Database futures, Course review	Relational algebra	Sample exam	-	Quiz 6	-

Resources for Students

Any of the following books is a suitable textbook for this course:

- Fundamentals of Database Systems (http://wps.aw.com/aw_elmasri_fundatasys_7) , Elmasri and Navathe, 7th edition, 2016, Addison-Wesley
- Database System Concepts (http://db-book.com/) , Silberschatz, Korth, Sudarshan, 6th edition, 2010, McGraw-Hill
- Database Systems: The Complete Book (http://www-db.stanford.edu/~ullman/dscb.html) , Garcia-Molina, Ullman, Widom, 2nd edition, 2008, Prentice-Hall
- Database Systems: An Application-Oriented Approach (http://wps.aw.com/aw_kifer_datacomp_2) Kifer, Berstein, Lewis, 2nd edition (Complete Version), 2006, Addison-Wesley

Choose the one that best suits your learning style and preferences. Also, if you have access to an earlier edition of any of these books (one or two editions less than the one given above), it will be fine for this course.

The textbooks give the greatest detail on the topics covered in the course, but cover many other topics. The Course Notes have less detail, but cover exactly the course syllabus. The Lecture Slides have even less detail, but include the examples discussed in lectures.

The software systems to be used in this course are PostgreSQL, SQLite, Python and psycopg2. The documentation and manuals provided with PostgreSQL (https://www.postgresql.org/docs/11/index.html) and Python (https://docs.python.org/3.7/) are actually very good, and the SQLite (http://www.sqlite.org/docs.html) and psycopg2 (http://initd.org/psycopg/docs/) documentation is also reasonable, so you don't need to buy textbooks for these. However, if you feel more comfortable with a book, there are references to a range of books on the web sites for PostgreSQL (http://www.postgresql.org/docs/books/) , SQLite (http://www.sqlite.org/books.html) , and Python (https://www.learnpython.org/) .

A general problem with technology textbooks is that they go out-of-date very quickly. Another problem is that many of them provide a brief introduction with some examples, and then give a summary of the manual. In general, we have found O'Reilly books tend to be better than most.

PostgreSQL will be used as the primary DBMS in this course because it is a typical example of a full-featured client-server DBMS, and has the added bonuses that (a) it has a simple extensibility model and (b) has the source code available if you want to learn more about how DBMSs work. Commercial alternatives could have been Oracle, DB2, SQL Server, although none of these are available in source code form. (MS Access is not a full-featured relational database.) The only plausible open-source alternative is MySQL, but it was not a full-featured DBMS until version 5, and has a source code base that is largely cobbled together from a number of existing open-source systems. The PostgreSQL code base, on the other hand, is the result of coherent development by a relatively small team (although the developer base has expanded in recent years).

SQLite is a very widely-used example of a serverless RDBMS and will be used for contrast with PostgreSQL. It has a similar (90% compatible) dialect of SQL to PostgreSQL, and, like PostgreSQL, supports the SQL92 standard. As an exercise in portability, we will be expecting whatever SQL statements you write for this course to work on both PostgreSQL and SQLite (although this may not always be possible).

For programming interaction with the database, we will be using Python via psycopg2.

Course Evaluation and Development

This course is evaluated each session using the myExperience system. In the previous offering of this courses, students noted that the assignment is a little bit complicated, and some specification is not clear enough. We will refine the documents and simplify the specification in this term.

Resource created 2 months ago (Saturday 29 January 2022, 03:52:08 PM), last modified 25 days ago (Friday 25 February 2022, 05:26:34 PM).

Comments

(/COMP3311/22T1/forums/search?forum_choice=resource/70680)

(/COMP3311/22T1/forums/resource/70680)

Add a comment

There are no comments yet.

Course Outline

Contents

Course Details

Course Code	COMP1521
Course Title	Computer Systems Fundamentals
Units of Credit	6
Course Contact	**<cs1521 at cse.unsw.edu.au>**
Lecturer	Andrew Taylor <andrewt at unsw.edu.au>
Administrator	Dylan Brotherston <d.brotherston at unsw.edu.au>
Administrator	Zac Kologlu <z.kologlu at unsw.edu.au>
Classes	**Lectures**: 1UGA: Mon 14:00—16:00; Thu 15:00—17:00; ... timetable for all classes
Course Website	https://cgi.cse.unsw.edu.au/~cs1521/22T1/
Course Forum	https://edstem.org/au/courses/7762/discussion/
Handbook Entry	https://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1521/

Course Summary

This course introduces students to how computer systems are structured in terms of basic electronic components, how they are used to implement procedural programs, and how they are structured as a collection of software layers. It introduces students to low-level software layers such as operating systems and network infrastructure, and introduces concurrency concepts. The goal is to give students a solid understanding of what happens when high-level programs are executed, as a basis for further study in important areas of computing such as computer architecture, operating systems.

Assumed Knowledge

Before commencing this course, students should be able to ...

- write simple programs in the C programming language
- define and invoke functions and return results in C
- define and manipulate structured data in C
- use pointers to access data objects

These are assumed to have been acquired in COMP1511 or COMP1911.

Learning Outcomes

After completing this course, students will be able to ...

1. Describe the architectural layers (fundamental parts) of a modern computer systems from hardware device (chip) levels upwards
2. Describe the principles of memory management and explain the workings of a system with virtual memory management
3. Explain how the major components of a CPU work together, including how data and instructions are represented in a computer
4. Design, implement, and analyse small programs at the assembly/machine level
5. Describe the relationship between a high-level procedural language (C) and assembly (machine language) which implements it, including how a compiled program is executed in a classical von Neumann machine
6. Explain how input/output operations are implemented, and describe some basic I/O devices
7. Describe the components comprising, and the services offered by, an operating system
8. Implement simple programs involving communication and concurrency

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

Graduate Capability	**Acquired in**
scholarship: understanding of their discipline in its interdisciplinary context	lectures, assignments
scholarship: capable of independent and collaborative enquiry	lab work, assignments
scholarship: rigorous in their analysis, critique, and reflection	tutorials
scholarship: able to apply their knowledge and skills to solving problems	tutorials, lab work, assignments
scholarship: ethical practitioners	all course-work, by doing it yourself
scholarship: capable of effective communication	blog, tutorials
scholarship: digitally literate	everywhere in CSE
leadership: enterprising, innovative and creative	assignments
leadership: collaborative team workers	lab work, assignments
professionalism: capable of operating within an agreed Code of Practice	all prac work

Teaching Strategies and Rationale

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

- Lectures ... introduce concepts, show examples
- Tutorials ... reinforce concepts and provide additional examples
- Lab Work ... provide examples of using various technologies
- Assignments ... allow you to solve larger problems

Having said that, the second half of the course is more discursive than other CSE foundational courses.

This course is taught the way it is because it aims to give a broad view of many topics in computer systems, to provide a foundation for further study in later systems-related courses. At the same time, it provides further practice in developing software, but at a level closer to the machine than other foundational courses.

Lectures

Lectures will be used to present the theory and practice of the techniques and tools in this course. The lectures will include practical demonstrations of various technologies. Lecture notes will be available on the course web pages before each lecture.

All lectures will be online-only, delivered via YouTube. During lectures, discussion and question are welcome via YouTube's live event chat.

About fifteen minutes prior to each lecture, an announcement will be sent to the class via the course forum (and thus by email) which will include links to join the live lecture.

Recording of all lectures will be made available. Lectures may be pre-recorded on some topics.

Tutorials

From week 1, you will also be expected to participate a one-hour tutorial session to 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 Web in the week before each tutorial. There are no marks for tutorial participation.

Most tutorial classes (and laboratories) will run online via Blackboard Collaborate. A link on Moodle will take you to Blackboard Collaborate, where you will find a list of tutorial classes.

Some tutorial and laboratory classes are running face-to-face. If you have enrolled in such a class, please ensure you follow UNSW's Safe Return to Campus guidance.

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 involving the tools introduced in lectures. Because this course has a significant practical component, laboratory classes are **important**. If you do not put a good amount of effort into the lab classes, you risk failing the final exam.

Each week, there will be one or more exercises to work on. These exercises will be released in the week preceding the lab class.

During the lab, your tutor will provide feedback on your approach to the problem and on the style of your solution. Some labs may contain exercises which will be assessed during the lab.

Completed exercises need to be submitted. You must submit exercises before the deadline using **give** to obtain a mark for a lab exercise, The usual lab exercise submission deadline will be 20:59 Monday some lab exercises may have an extended deadline

The lab exercises for each week are worth in total 2 marks. All of your lab marks will be summed to give you a mark out of 18; if their sum exceeds 15, your total mark will be capped at 15.

Most labs include one or more challenge exercises. Challenge exercises may involve concepts not covered in lectures and they range in difficulty from not-very-hard to almost-impossible.

The contribution of challenge exercises to lab marks will be limited to 20%; hence you can obtain nearly all (over 95%) marks available for the lab component without completing challenge exercises.

If you wish to obtained a high mark for COMP1521, attempting some challenge exercises is highly recommended.

If your goal is just to master the core material and pass COMP1521, you can ignore challenge exercises.

Assignments

There are two assessable programming assignments. Assignments give you the chance to practice what you have learnt on relatively large problems (compared to the small exercises in the labs). Assignments are a *very important* part of this course, therefore it is essential that you attempt them yourself.

- Assignment 1, on Assembly programming (MIPS); due Week 7; worth 15%
- Assignment 2, on System programming; due Week 10; worth 15%

Late assignments submissions will be penalised. The exact penalty will be specified in the assignment specification: often, it is a 1% reduction in the maximum achievable mark for every hour late.

Weekly Tests

There will be weekly tests from weeks 3–10 designed to give you timely and realistic feedback of your understanding of the course material. Tests may be programming exercises, multiple choice questions, or both.

These will be conducted in your own time under self-enforced exam-like conditions. Each test will specify the conditions, but typically these will include:

1. no assistance permitted from any person;
2. a time limit;
3. no access to materials (written or online) except specified language documentation or man pages.

Each test is worth 1.7 marks, and will be automarked. Your total mark for the tests component is computed as a sum of your best 6 of 8 test marks. Any violation of the test conditions will result in a mark of zero for the entire test component.

Final Exam

There will be an online exam which students completely remotely (from home). This will be centrally timetabled, and appear in your UNSW exam timetable.

During this exam you will need to execute, debug and test your answers to implementation tasks which will be similar to those encountered in lab exercises and weekly tests.

There is a hurdle requirement on the final exam. If you do not score at least 40% (18.0/45) on the exam (after scaling), you cannot pass this course. If your overall course score exceeds 50%, despite scoring very poorly (<40%) on the exam, the hurdle will be enforced via a grade of UF.

Student Conduct and Academic Integrity

Student Conduct

The **Student Code of Conduct** (Information, Policy) 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.

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, in 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.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer, Grievance Officer, or one of the student representatives.

Academic Integrity

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 is defined as using the words or ideas of others and presenting them as your own. Plagiarism undermines academic integrity, and is not tolerated at UNSW. Instances of plagiarism are treated by UNSW and CSE as acts of academic misconduct, which carry penalties as severe as being excluded from further study at UNSW. There are several on-line resources to help you understand what plagiarism is and how it is dealt with at UNSW.

- Plagiarism and Academic Integrity
- UNSW Plagiarism Procedure

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, at CSE you are responsible for ensuring that your assignment files are not accessible by anyone but you by setting correct permissions in your CSE home directory and for any code repositories you may use. 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.

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

- Student Code Policy
- Student Misconduct Procedure
- Plagiarism Policy Statement
- Plagiarism Procedure

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

- Essential Advice for CSE Students

Assessment

Item	Topics	Due	Marks	LOs
Tests	all topics	Weeks 3-10	10	1-9
Assignment 1	Assembly programming (MIPS)	Week 7	15	4
Assignment 2	System programming	Week 10	15	2,5
Labs	all topics	Week 1-10	15	1-5,9
Final Exam	all topics	exam period	45	1-9

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

CourseWorkMark	=	TestMark + LabMark + Ass1Mark + Ass2Mark	out of 55
ExamMark			out of 45
ExamOK	=	ExamMark ≥ 18.0/45	true/false
FinalMark	=	CourseWorkMark + ExamMark	out of 100
FinalGrade	=	UF, if ! ExamOK && FinalMark ≥ 50 FL, if FinalMark < 50/100 PS, if 50/100 ≤ FinalMark < 65/100 CR, if 65/100 ≤ FinalMark < 75/100 DN, if 75/100 ≤ FinalMark < 85/100 HD, if FinalMark ≥ 85/100	

Course Schedule

The following is a rough schedule of when topics will be covered. This will most likely change over the session as topics take more or less time to cover.

Week Week	Lectures Lectures	Tut/Lab Tut/Lab	Assigns Assigns	Tests Tests
1	course intro; MIPS assembly programming	C revision	-	-
2	MIPS assembly programming	MIPS assembly programming	-	-
3	MIPS assembly programming	MIPS assembly programming	-	test 1 released (due week 4)
4	MIPS assembly programming	MIPS assembly programming	-	test 2 released (due week 5)
5	Bit manipulations; Integer representations	MIPS assembly programming	-	test 3 released (due week 7)
6	flexibility week	-	-	test 4 released (due week 7)
7	IEEE-754; UTF-8	Bit manipulations; Integer representations	assign 1 due	test 5 released (due week 8)
8	file metadata; file systems	IEEE-754; UTF-8	-	test 6 released (due week 9)
9	virtual memory; processes; signals	manipulating files, metadata, and directories	-	test 7 released (due week 10)
10	parallelism; synchronisation; coordination; communication	virtual memory; processes; signals	assign 2 due	test 8 released (due week 11)

Resources for Students

There is no single 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 lecture notes should provide sufficient detail to introduce topics, and you will then study them in further depth in the tutes, labs and assignments.

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.

The following is a Recommended Reading for this course:

- *Computer Systems: A Programmer's Perspective*, by Randal E. Bryant and David R. O'Hallaron; Prentice-Hall (web site)

There are copies in the UNSW Bookstore and in the library. It covers many of the topics in the course, but uses a different machine architecture (i.e., not MIPS).

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

- *Introduction to Computer Systems: From Bits and Gates to C and Beyond*, by Yale N. Patt and Sanjay J. Patel; McGraw Hill
- *The Elements of Computing Systems: Building a Modern Computer from First Principles*, by Noam Nisan and Shimon Schocken; MIT Press (web site, including lecture slides)

Documentation for the various systems used in the course is linked from the course website.

Course Evaluation and Development

Every term, student feedback is requested in a survey using UNSW's myExperience online survey system where the feedback will be used to make improvements to the course.

Students are also encouraged to provide informal feedback during the session, and to let course staff know of any problems as soon as they arise. Suggestions will be listened to openly, positively, constructively, and thankfully, and every reasonable effort will be made to address them.

This course has previously run in 17s2, 18s1, 18s2, 19T2, 19T3, 20T2, 20T3, 21T2, 21T3.

We have read carefully the myExperience feedback from 21T3 and in response have made changes to the delivery:

- We have moved the MIPS material earlier in the course, to allow students to start assignment 1 earlier.
- We have also made minor changes to the ordering and emphasis of other material so it is more coherent.

Resources / Course Outline

Course Outline

🎉 🎉 Welcome to COMP1531 🎉 🎉

COMP1531 is the first station on a very long voyage to becoming the best software engineer you can be. We're going to have a great term - even if we're all not physically together. COMP1531 has a teaching team of 35 passionate teaching staff - nearly all of whom have been in your shoes within the last few years. Everyone here will primarily interact with two tutors throughout this term. We're all excited to meet you.

Through the term you'll all working not only with your tutor(s), but also with our course admins and lecturers.

Emily Luxa (https://webcms3.cse.unsw.edu.au/users/z5258919) 🍍

Hayden Smith (https://webcms3.cse.unsw.edu.au/users/z3418003) 🍇

This page offers an outline of this course. Take the time to read it, as it covers everything that we expect you from this term, and everything you should expect from us!

- COMP1531 teaching team

Contents

- 8. Course Evaluation and Development

1. Course Details

Find information relating COVID-19 and this offering here (https://www.covid-19.unsw.edu.au/safe-return-campus-faqs) and here (https://edtech.eng.unsw.edu.au/c19mess/comms.html) .

Course Code	COMP1531
Course Title	Software Engineering Fundamentals
Convenor	Hayden (https://webcms3.cse.unsw.edu.au/users/z3418003)
Lecturer(s)	Hayden (https://webcms3.cse.unsw.edu.au/users/z3418003) , Jake (https://webcms3.cse.unsw.edu.au/users/z3534499)
Admin(s)	Nick (https://webcms3.cse.unsw.edu.au/users/z5169779) , Emily (https://webcms3.cse.unsw.edu.au/users/z5258919)
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs1531/21T3/ (http://cse.unsw.edu.au/~cs1531/21T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1531.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1531.html)

2. 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.

You will be exposed to agile software practices, team collaboration and effective communication through implementing a group project based on agile software methodologies that requires you to analyse, design, build and deploy a web-based application. This course is typically taken in soon after completing COMP1511, but could be delayed and taken later. It provides essential background for the teamwork and project management required in many later courses.

2.1. 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.

2.2. Student Learning Outcomes

After completing this course, students will be able to:

1. Demonstrate effective use of applying the python programming language to solve problems in relation to web-based applications
2. Demonstrate proficiency in use of system schematics, data modelling, and state modelling, to analyse complex software systems
3. Identify the complexities of software design and development, including design smells and common best design practices
4. Demonstrate effective usage of testing methods (e.g., unit tests, integration tests, test plan/cases, test automation)
5. Understand the software engineering life cycle and agile software development practices, to elicit requirements, design, implement, and iterate on software projects.
6. Understand key characteristics of a functioning team, both in terms of understanding professional expectations, and the use of git as a source code management tool for collaboration.

---

3. Teaching Strategies

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

- Lectures
- Tutorials
- Laboratories
- Help Sessions
- Major Group Project
- Final Exam

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.

3.1. 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.

3.2 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.

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.

Tutorial information on it's release can be found by going to the tutorial page (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60502) . Tutorial solutions will be released at the end of the week the tutorial is in.

Tutorials contribute to your class mark.

Tutorials will be run via zoom. Zoom is chosen in preference to Blackboard Collaborate due to the relatively higher degree of reliability zoom 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 COMP1531 is operated.

3.3. Laboratories

In terms of the weekly workload, each week with a tute/lab you will be given a "lab" (i.e. take-home activities to complete in your own time) that is due at the end of that week.

In terms of the 2 hour period each week allocated for labs, these sessions run immediately after your tutorial. For any given class, your tutor and lab assistant will run two separate video calls at the same time during the 2 hours.

- **(1) Project check-ins** . Your tutor will run a 2 hour session where each group will be scheduled 20 minutes for a project check in.
- **(2) Marking labs off** . Your lab assist will run a 2 hour session where each group will be scheduled 15 minutes for them to mark your labs off and answer questions about labs. Your tutor will mark your lab off with your other group members in the "room", so be prepared in advance that your group members will be there for part of the marking.

A schedule of what times your group should attend which sections can be found on the tutorial and labs schedule page (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60518) .

Lab solutions will be released 1 week after the end of the week they are due (with exception of first few weeks where lab solution releases will be delayed due to late enrollments).

Appropriating responding to, and actioning any feedback given by tutors in previous labs is expected by all students.

Labs contribute to your class mark.

3.3.1. (1) Project check-ins

In weeks 1, 2, 3, 5, 8, and 9, your project check ins are a chance your group and your tutor to sync up about your progress and work through problems together.
In weeks 4, 7, and 10, your project check in periods will consist of a presentation and Q&A as specified in the project specification.

3.3.2. (2) Marking labs off

Labs will typically be due at the end of the week they are provided. You are required to show & discuss your lab with your lab in the lab of the week immediately following the week the lab is due. Submitting the lab is not enough to be awarded a mark. You cannot obtain marks by emailing your tutor your labs. If you are unable to submit your lab on time due to extenuating circumstances, please apply for special consideration.

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.

The timetable for help sessions can be found here (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60519) .

3.5. Major Group Project

There will one main group project which will run through the teaching period from weeks 1-10 and contributes to **50%** of the overall course mark.

The course staff will form groups of 4-5 students in your allocated tutorials at the beginning of the course. You will be notified of what group you're in during your week 1 tutorial & lab. A short survey will be sent no later than the 8th of September 2021 that will allow you to make preferences of who is in your group. These groups will and 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 milestones. Each milestone 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 milestone, you (as a team) will demonstrate to your lab class the functionality implemented during that milestone. 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 milestone. You will need to carefully design the solution for your current milestone, such that the solution is extensible to accommodate these changes.

After each milestone due date, your group will present your work in the next lab that occurs. This is outlined in the major project specification. To receive a mark for that milestone, each team member must be present for the demonstration during the relevant lab time, with working audio and with their camera on. Having no audio or video will result in a loss of marks. If you are unable to attend, you must apply for special consideration and have your application accepted.

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

3.6. Final Exam

There will be a centrally timetabled final exam which will in your UNSW exam timetable. The exam may contain a mixture of short answer questions and programming exercises. More specific details of the exam will be provided through the course.

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.

---

4. Assessment

Item	Due	Weighting
Class Mark (Tutorials + Labs)	Weeks 1,2,3,4,5,7,8,9	15%
Major Project	Milestones due beginning of weeks 4, 7, 10	50%
Final Exam	Exam period	35%

4.1. Class Mark (15%)

Your class mark is made up of marks associated with tutorials, and marks associated with labs. There is a total of 18 marks that can be gained between tutes and labs throughout the course (i.e. 3 bonus marks), although it will be capped at 15 overall.

- **Your Labs** mark is out of 10. Each **lab** you submit is worth up to 1.25% of the course. Each lab has a series of questions, where each question has a number of points associated with it. Each lab has a target number of points (e.g. 8) where if you score 8 points in that lab you receive the full 1.25%. This means for any given lab you only need to complete a subset of the questions satisfactorily to get full marks. This varies per week - lab instructions have the full info. If you score more points than the maximum for that lab, you will not receive bonus marks beyond that lab.
- **Your Tutorial** is out of 8 marks. attendance and participation in tutorials 1,2,3,4,5,7,8,9 will add up to 8% of the course. 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 challenge 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.

Labs have no late penalty , **because late submissions are not accepted** . While this may seem harsh, some courses have their labs due on the previous Friday - so we are giving you the full weekend to complete your lab. The reason we have no late penalties is because we have to run automarking and provide feedback to you between your lab submission and lab the following week, and therefore once the labs are submitted and marked, solutions will be known to all students.

Please don't ask for bonus marks for your labs - you can get 100% in your class mark without ever doing any challenge exercises, so there are bonus marks implicitly built in.

4.2. Major Project (50%)

The marking criteria for your major project will be specified in the major project specification which can be (once released) found here (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60521) .

4.3. Final Exam (35%)

The marking criteria for your final exam will be specified in the exam specification which can be (once released) found here (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60736) .

---

5. Course Schedule / Timetable

The schedule for this course is outlined clearly in the timetable for lectures (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60517) , tutorials/labs (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60518) , and help sessions (https://webcms3.cse.unsw.edu.au/COMP1531/21T3/resources/60519) .

---

6. 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)

At any time in this course, when you push code to gitlab from your own machine (locally or on CSE account), you are acknowledging that this code you push is your own work, except here permitted by the originality rules for this course and this assessment. You acknowledge that this pushed code has not been submitted for academic credit elsewhere, and you acknowledge that you have read an understood the University Rules in respect of Student Academic Misconduct.

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 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. There is also a section on Webcms3 that

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. Anonymous feedback forms will be provided to students throughout the course.

The main improvements to be made from the previous offering are to include reducing the impact and burden of labs, having more consistent peer reviewing throughout the course, and improving on the quality of some lecture resources.

This is being addressed during the 21T3 offering.

Resource created 10 months ago (Saturday 15 May 2021, 11:20:58 PM), last modified 6 months ago (Tuesday 21 September 2021, 04:33:34 PM).

Comments

(/COMP1531/21T3/forums/search?forum_choice=resource/60500)

(/COMP1531/21T3/forums/resource/60500)

There are no comments yet.

Course Outline

Contents

Course Details

Course Code	COMP3331/9331
Course Title	Computer Networks and Applications
Units of Credit	6
Lecture	Salil Kanhere
Admin	Ayda Valinezhad Orang
Classes	Lectures: Mon 13:00-15:00 Hrs, Tue 16:00 -18:00 Hrs, On-line: Zoom Meeting (links on Lectures page) Timetable (https://webcms3.cse.unsw.edu.au/COMP3331/20T3/timetable) for all classes.
Consultations	TBA Venue: on-line
Course Website	Course Website (https://webcms3.cse.unsw.edu.au/COMP3331/20T3/)
Course Contact Email	cs3331@cse.unsw.edu.au (mailto:cs3331@cse.unsw.edu.au)
Handbook Entry	Handbook Entry (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3331.html)

Course Summary

This course is an introductory course on computer networks, aimed at students with a computer science / electrical engineering background. We will focus on common paradigms and protocols used in present data communication. Through lectures, in-class activities, labs and assignments, you will learn the theory and application of:

(1) Medium access control, congestion control, flow control, and reliable transmission,

(2) Addressing and naming,

(3) Routing and switching,

(4) Widely used protocols such as Ethernet, IP, TCP, UDP, HTTP, etc.

(5) Security threats and common defensive techniques, and

(6) Special-purpose networks such as content delivery networks, peer-to-peer networks and wireless networks.

This is a combined undergraduate and postgraduate course.

Course Timetable

There will be 4 hours of lectures every week:

(i) 2-hour lecture on Monday 13:00 - 15:00 and

(ii) 2-hour lecture on Tuesday 16:00 - 18:00

We plan to run live online lectures through Zoom with the opportunity to ask questions via chat. Recording of all lectures will be made available. Links for the Zoom meetings will be posted on the Lectures page.

There will be 2-hour labs during 8 weeks (starting in Week 2). The detailed lab schedule will be posted on The detailed course timetable is available here (https://webcms3.cse.unsw.edu.au/COMP3331/21T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)

(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)

(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable)

(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)
(/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable) (/COMP3331/19T2/timetable)

Course Aims

- To provide an in-depth introduction to a wide range of topics in the field of computer networks including the Internet.
- To get a hands-on understanding of the working of network protocols.
- To gain expertise in network programming, designing and implementing network protocols, evaluating network performance and problem-solving skills.
- To build the necessary foundational knowledge required in subsequent networking courses (COMP4335-4337, COMP6733, COMP9332-9337).

Student Learning Outcomes

After completing this course, students will:

- Have a working knowledge of computer networks, and will be able to demonstrate by describing aspects of the topics and by solving problems related to the topics
- Have a solid understanding of the current architecture of the Internet and the entities involved in its operations
- Be able to identify soundness or potential flaws in proposed protocols
- Be equipped with the necessary skills to design networked applications and protocols
- Implement and write protocols and applications in C, Java or Python
- Analyze and evaluate the performance of computer networks
- Be able to capture and network traffic
- Be able to understand and explain security and ethical issues in computer networking

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

Graduate Capability	Acquired in
Scholarship: of their discipline in its interdisciplinary context	Lectures, labs, assignment
Scholarship: Capable of independent and collaborative	Labs, assignment
Scholarship: rigorous in their analysis, critique, and reflection	Lectures, labs, exams, sample problems
Scholarship: able to apply their knowledge and skills to solving problems	Labs, assignment, exams, sample problems
Scholarship: capable of effective communication	Labs, assignment, lectures, exams
Scholarship: digitally literate	All aspects of the course
Scholarship: information literate	All aspects of the course
Leadership: collaborative team workers	Labs, assignment
Professionalism: capable of independent, self-directed practice	All aspects of the course
Professionalism: capable of lifelong learning	All aspects of the course
Professionalism: capable of operating within an agreed Code of Practice	Labs, assignment
Global citizens: culturally aware and capable of respecting diversity and acting in socially /responsible ways	Labs, course forums

Assumed Knowledge

Before commencing this course, students should:

- **Have a good understanding of data structures and algorithms, basic probability theory.**
- **Be able to write working programs in C, Java or Python. The course will include programming assignment and labs.**

Teaching Rationale

This course takes a top-down approach to teaching computer networks. The rationale behind this is that most students have first-hand experience using applications running over the Internet. This allows them to relate to each layer of the stack as we travel down the layers. Once they are committed, they participate in appropriate

cognitive aspects such as learning the details with a focus to understand them. Students get mentally prepared to answer questions. Very often there is no single answer or the answers can be unexpected. This results in deep learning and gives students a sense of accomplishment and confidence.

Learning will be largely facilitated through the delivery of lectures. The hands-on laboratories will provide an opportunity to gain a deeper understanding of the concepts discussed in the lectures. The sample problems, homework problem set, and tutorials will help develop problem-solving skills and prepare for the exams. The programming assignments are mainly geared to allow students to gain familiarity with basic network programming and designing network protocols.

Teaching Strategies

- **Lectures:** introduce theory demonstrate how they apply in practice
- **Lab Work:** reinforce concepts taught in lectures by conducting hands-on experiments and network performance
- **Assignment:** allow students to design and implement network evaluate network performance
- **Homework Problems:** allow students to solve problems based on content from lectures, develop problem-solving skills, assist with exam preparation
- **Consultations, Tutorials and Course Forum:** allow students an opportunity to ask questions and seek help.

Assessment

There will be four assessment components as listed below:

Component	Weight
Lab Exercises	20%
Programming Assignment	20%
Mid-term Test	20%
Final Exam	40%

To pass the course a student MUST receive at least 40% marks in the final exam. The following formula outlines precisely how the final mark will be computed:

```
lab = marks for lab exercises (scaled to 20)
assign = marks for the programming assignment (out of 20 marks)
midTerm = mark for the mid-term exam (out of 20 marks)
finalExamScaled = scaled mark for the final exam (out of 40 marks)
mark = lab + assign + midTerm + finalExamScaled
grade = HD|DN|CR|PS if mark >= 50 && finalExamScaled >= 16
      = FL          if mark < 50
<p>      = UF          finalExamScaled < 16</p>
```

In this course, the final and mid-term exam will be using the **Inspera platform** which you will get access to from a link in the course Moodle site. More information for students about Inspera can be found here: https://unsw.sharepoint.com/sites/Assessment-Platform-Pilot (https://unsw.sharepoint.com/sites/Assessment-Platform-Pilot) .

Plagiarism is defined as (https://student.unsw.edu.au/what-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 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)
- 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 (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)
- 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)

Make sure that you read and understand the above. Ignorance is not accepted as an excuse for plagiarism. In particular, you are responsible for securely storing your assignment files such that they are not accessible by anyone but you by setting proper permissions on your CSE home directory and/or on online code repositories. 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.

Course Schedule

The following table lists the tentative weekly schedule. Students will be informed of any changes during the lecture and by announcements on the notices page.

Week	Lecture Dates	Lecture Topics	Labs	Assessment Tasks
1	31 May & 1 June	Course Logistics Introduction: • What is the Internet? • Network edge and core • Performance of networks Introduction: • Layering and Encapsulation	Self-Study of Lab resources and tools (no submission and no marks)	

Week	Lecture Dates	Lecture Topics	Labs	Assessment Tasks
2	7 & 8 June	Application Layer: • Principles of networked applications • The Web & HTTP • Email • Domain Name Service (DNS) • Peer-to-Peer Networks and DHT	Lab 1	Lab 1 submission deadline: 16:00 Tue 15 June
3	**14** & 15 June NOTE: 14th June is a **public holiday** . A recorded lecture will be available for viewing	Application Layer: • Content Distribution Networks • Socket Programming Transport Layer: • Transport services	Lab 2	Lab 2 submission deadline: 16:00 Tue 22 June **Assignment Specs Released (Expected)**
4	21 & 22 June	Transport Layer: • Multiplexing & Demultiplexing • UDP • Principles of reliable data delivery • Pipelined Protocols	Lab 3	Lab 3 submission deadline: 16:00 Tue 29 June
5	28 & 29 June	Transport Layer: • TCP • Connection management & flow control • Congestion control • Fairness	Tutorial 1	
6	No lecture	No lecture	No Lab	No Lab

Week	Lecture Dates	Lecture Topics	Labs	Assessment Tasks
7	12 & 13 July	Network Layer, Data Plane: • Overview • IP	Lab 4	**Mid-term Test on 12th July during lecture hours** Lab 4 submission deadline: 16:00 Tue 20 July
8	19 & 20 July	Network Layer, Data Plane: • IP Addressing • NAT • IPv6	Lab 5	Lab 5 submission deadline: 16:00 Tue 27 July
9	26 & 27 July	Network Layer, Control Plane: • Routing algorithms • Link State and Distance Vector • Hierarchical routing • ICMP Link Layer: • Error detection	Lab 6	Lab 6 submission deadline: 16:00 Tue 3 August

Week	Lecture Dates	Lecture Topics	Labs	Assessment Tasks
10	2 & 3 August	Link Layer: • Multiple Access Protocols • Link Layer Addressing and ARP • Ethernet • Switches Wireless Networks • Wireless characteristics • 802.11 • CSMA/CA A day in the life of a web request Network Security • Basic Cryptography • Message integrity & Digital signatures • Authentication	Tutorial 2	**Assignment Due**
Exam Period	13 - 26 Aug	T2 Exams		**Final Exam**

Resources for Students

Course Textbook:

- Computer Networking - A Top-Down Approach Featuring the Internet, J. Kurose and K. Ross, Pearson, 7th Edition, 2017 (Sixth edition will suffice for most parts). Link for ordering e-book: Pearson (https://www.pearson.com.au/9781292153605#!)

Reference Texts:

- Unix Network 1 - Networking APIs: Sockets and XTI, W. Richard Stevens, Prentice Hall, Second Edition, 1998.
- Java Network Programming, E. R. Harold, O'Reilly, Third Edition, 2004.
- Learning Python, Mark Lutz, O'Reilly, Fifth Edition, 2013.
- Computer Networks: A Systems Approach, Larry Peterson and Bruce Davie, Morgan Kaufmann, Fifth Edition, 2011.
- Introduction to Computer Networks and Cybersecurity, John Wu and J. David Irwin, CRC Press, 2013.
- Computer Networks, Andrew Tanenbaum and David Wetherall, Fifth Edition, Pearson, 2010.

Links to additional reading material will be available on the lecture notes page.

Software:

For the labs, we will be using several Unix-based network utility programs. The purpose of these programs and information on how to use them will be provided in the lab handouts. We will also use a packet sniffing tool called Wireshark which has been widely deployed on CSE machines. In addition, we will also use Ns-2 a widely used network simulator for a few labs. Ns-2 is installed on the CSE lab machines. The simulator is written in C++. However, it uses OTcl (https://en.wikipedia.org/wiki/OTcl) as its command and configuration interface. In the lab exercises, we will use scripts written in OTcl. We will provide the necessary scripts for the lab exercises. You will be expected to run the scripts, make some changes in the scripts, and certain performance metrics. You will not be required to write C++ code. Detailed resources for all tools used will be made available on the lab exercises page.

The programming assignment is expected to be developed in **C, Java or Python** . Students are assumed to have sufficient expertise in one of these programming languages. Links to network programming resources in C, Java and Python will be available on the assignment page. Students will work on some simple client-server applications in two lab exercises which should be a useful starting point for the assignment.

Course Evaluation and Development

Student feedback on this course and the effectiveness of lectures in this course are obtained via the myExperience survey at the end of each term. Student feedback is taken seriously, and continual improvements are made to the course based in part on this feedback. Students are strongly encouraged to let the lecturer in charge know of any problems as soon as they arise. Suggestions and criticisms will be listened to openly, and every action will be taken to correct any issue or improve the students' learning experience. Some specific changes I will be incorporating this term based on the feedback I received in 2020 Term 3:

- Check the chat stream more frequently during live lecture for questions
- Using a different course forum than WebCMS3. We will use Ed this term
- Prepare some short videos focused on problem solving to help students prepare for the exams

Special Consideration

You can view the Special Consideration policy at the link here (https://student.unsw.edu.au/special-consideration) . (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-

consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration) (https://student.unsw.edu.au/special-
consideration) (https://student.unsw.edu.au/special-consideration)

UNSW handles special centrally (in the Student Lifecycle division), so all special must be submitted via the UNSW Special Consideration (https://iaro.online.unsw.edu.au/special_consideration/home.login) website. If your work in this course is affected by unforeseen adverse circumstances, you should apply for Special Consideration. Special consideration requests must be accompanied by documentation on how you have been affected, which will be verified by Student Lifecycle. Do not email the LiC directly about special consideration. If your request is reasonable and your work has clearly been impacted, then

- an assignment, you may be granted an extension
- Mid Term Exam, you may be granted an opportunity to take the exam later
- Final Exam, you may be offered a Supplementary Exam

Note the use of the word "may". None of the above is guaranteed. It depends on you making a convincing case that the circumstances have clearly impacted your ability to work. Note that UNSW expects you to be available to sit Supplementary Exams if required. If you are awarded a supplementary exam and do not attend, then your exam mark will be zero.

If you are registered with Disability Services, please forward your documentation to your Lecturer within the first two weeks of term.

Contacting LiC and Course Admin: No personal emails, please. Please use cs3331@cse.unsw.ed.au. (mailto:cs3331@cse.unsw.edu.au)

9 months ago (9 months ago) , last modified 8 months ago (8 months ago) .

Resource created 11 months ago (Sunday 09 May 2021, 09:14:58 PM), last modified 6 months ago (Friday 10 September 2021, 05:34:06 PM).

Course Outline

Contents

Course Details

Course Code	COMP3900/COMP9900
Course Title	Computer Science / Information Technology Project
Lecturer in Charge	Matthew Sladescu (https://webcms3.cse.unsw.edu.au/users/z3531589)
Tutors/Mentors	Rachid Hamadi (https://webcms3.cse.unsw.edu.au/users/z2286838) Iwan Budiman (https://webcms3.cse.unsw.edu.au/users/z3532588) Ali Darejeh (https://webcms3.cse.unsw.edu.au/users/z5081642) Xin Li (https://webcms3.cse.unsw.edu.au/users/z5203513) Dominic Wong (https://webcms3.cse.unsw.edu.au/users/z5208437) Lina Zhang (https://webcms3.cse.unsw.edu.au/users/z5100784) Armin Chitizadeh (https://webcms3.cse.unsw.edu.au/users/z3323393) Dong Luo (https://webcms3.cse.unsw.edu.au/users/z5165343)

Classes	**Lectures** : Monday 9:00am - 11:00am online in Weeks 1, 2, and 10, Tuesday 8am – 10am online in Week 3 Lectures in 2021 T2 will be delivered online through echo360, which will be accessible through Moodle. Timetable for all classes (/COMP9900/21T2/timetable) **Labs:** Weekly labs according to your timetable. Timetable for all classes (https://webcms3.cse.unsw.edu.au/COMP9900/21T2/timetable) Labs will be conducted online through Blackboard Collaborate Ultra, which will be accessible through Moodle.
Consultations	Contact Lecturer in Charge to arrange for a consultation
Units of Credit	6
Course Website	https://webcms3.cse.unsw.edu.au/COMP9900/21T2/ (https://webcms3.cse.unsw.edu.au/COMP9900/21T2/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3900.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3900.html) http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9900.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9900.html)

Course Summary

This is a software project capstone course. Students work in teams of ideally **five (5) members** to define, implement and evaluate a real-world software system. Most of the work in this course is team-based project work, although there are some introductory lectures on software project management and teamwork strategies. Project teams meet **weekly** starting from **Week 1** with project mentors to report on the progress of the project. Assessment is based on a project proposal, two progressive demonstrations and retrospectives, a final project demonstration and report, and on the quality of the software system itself. Students are also required to reflect on their work and to provide peer assessment of their team-mates' contributions to the project.

Assumed Knowledge

Before commencing this course, students should:

- have basic knowledge of database programming, Web programming and/or script programming (such as Python, PHP, and Javascript).
- be able to produce correct software programs in Python, Java or C/C++, i.e., compilation, running, testing, debugging, etc.
- be able to produce readable code with clear documentation.

Note:
For COMP9900, students must be in their final semester of study, and have completed at least 66 UOC towards MIT program 8543.
For COMP3900, students must have completed COMP1531, and COMP2521 or COMP1927, and are enrolled in a BSc Computer Science major with completion of 102 UOC.

Student Learning Outcomes

After successfully completing this course, students will:

1.be able to work from a set of requirements, elaborate them, and produce a specification
2.be able to design and build a correct, efficient and robust software system from specification
3.be able to use software development and software project management tools
4.be able to validate the correctness and robustness of software
5.be able to work effectively in a project team, and lead when required
6.be able to manage their time effectively, and make reasoned trade-offs over competing demands
7.be able to communicate technical information clearly, both verbally and in writing

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

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	Project
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Project
Professionals capable of ethical, self- directed practice and independent lifelong learning	Project
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Project

Teaching Strategies

- Lectures: the main way to introduce and overview software project management; and discuss various teamwork strategies as well as project management techniques.
- Labs: for the team to weekly meet up and interact with the mentor.
- Project: give students the hands-on experience on a real-world type software system, in a teamwork environment.

Teaching Rationale

The learning focus in this course is primarily a team-based software project (for students to practice their skills and knowledge in a real-world type software project, in a teamwork setup). The course will have an emphasis on the practical software development skills in a teamwork environment. Students will learn the basic concepts of software project management through introductory lectures.

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	Topic	Due	Marks	Contributes to
Proposal	Project	Monday, Week 4 @ 9:00am (21 Jun)	10%	CLOs 1,3,5-7
Progressive Demo A	Project	Week 5 Lab Time	2.5%	CLOs 2-7

Retrospective A	Project	Week 7 Lab Time	2.5%	CLOs 5
Progressive Demo B	Project	Week 8 Lab Time	2.5%	CLOs 2-7
Retrospective B	Project	Week 9 Lab Time	2.5%	CLOs 5
Software Quality	Project	Monday, Week 10 @ 10:00pm (2 Aug)	20%	CLOs 2-7
Project Report	Project	Monday, Week 10 @ 10:00pm (2 Aug)	20%	CLOs 1,2,5-7
Final Project Demo	Project	Week 10 Lab Time	20%	CLOs 2-7
Participation & Peer Assessment	Project	Friday Week 10 @ 10:00pm (6 Aug)	20%	CLOs 1-7

Course Schedule

The following table outlines a **provisional** schedule for this course. The contents of the lectures are described **roughly** and are subject to **adjustments.**

Week	Lectures	Labs	Assignments	Notes
1	Course introduction & Assessment Overview, User Stories	Introductory Lab activities		-
2	Agile, Scrum Framework, Using Jira for Scrum, Git & GitHub	Work on the project, progress report		-
3	Project Management techniques	Work on the project, progress report		-
4	-	Work on the project, progress report	Proposal due @ 9:00am, 21 June (Monday)	-
5	-	Work on the project, progressive **Demo A**	Progressive Demo A due in lab	-
6	-	-		-
7	-	Work on the project, progress report	Retrospective A due in lab	-
8	-	Work on the project, progressive **Demo B**	Progressive Demo B due in lab	-

9	-	Work on the project, progress report	Retrospective B due in lab	-
10	Final wrap-up lecture	Final project **demo**	**Project Report** and **Software Quality (Final System Code)** due @ 10:00pm on 2 August (Monday) **Final Project Demo** due in lab **Participation & Peer Assessment** due @ 10:00pm on 6 August (Friday)	-

Resources for Students

There are no specific texts and recommended readings for COMP3900/COMP9900. Programming language specific texts may be useful as references, depending on the programming language(s) used in the project. Other online resources and/or documentation related to this trimester's real-world projects will be provided in the course website during the trimester.

Safe Return to Campus FAQs (https://www.covid-19.unsw.edu.au/safe-return-campus-faqs)

UNSW Engineering Covid-19 Communication (https://edtech.eng.unsw.edu.au/c19mess/comms.html)

Student Reps (https://www.cse.unsw.edu.au/~stureps/)

Course Evaluation and Development

This course is evaluated each term using the myExperience system at the end of the term.

In a former offering of this course, some students expressed the desire to deliver their system on an alternative platform to vlab. To address this, in this term's offering, we will also provide an alternative option to vlab, as detailed in assessment guidelines.

Your feedback is important and will be considered to improve future offerings of this course. Students are also encouraged to provide informal feedback during the term, and let the lecturer in charge and mentors know of any problems as soon as they arise. Suggestions will be listened to very openly, positively, constructively and thankfully, and every reasonable effort will be made to address them as soon as possible.

Resource created 10 months ago (Sunday 23 May 2021, 12:41:03 PM), last modified 10 months ago (Sunday 23 May 2021, 05:06:52 PM).

Resources / Course Outline

Course Outline

Contents

Course Details

Course Code	COMP9444
Course Title	Neural Networks and Deep Learning
Convenor	Alan Blair (/users/z3029739)
Admin	TBA
Classes	Monday 2-4pm (Weeks 1-3, 5-10) and Thursday 6-8pm (Weeks 1-10) (no lecture on Monday of Week 4 due to the Labour Day Holiday)
Consultations	TBA
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs9444/19T3/ (http://cse.unsw.edu.au/~cs9444/19T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9444.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9444.html)

Course Summary

This course aims to introduce students to the main topics and methods in the field of neural networks and deep learning, ranging from traditional neural network models to the latest research and applications of deep learning.

Topics chosen from: perceptrons, feedforward neural networks, backpropagation, deep convolutional networks for image processing; geometric analysis of trained neural networks; recurrent networks, language processing, semantic analysis, long short term memory; Hopfield networks, restricted Boltzmann machines and

autoencoders, generative adversarial networks; deep reinforcement learning; designing successful applications of neural networks; recent developments in neural networks and deep learning.

Student Learning Outcomes

After completing COMP9444, students should

- understand aspects of the social, intellectual, and neurobiological context of neural networks and deep learning
- have an understanding of a variety of NN and DL techniques, including the Planned Topics listed below
- be able to analyse a problem for neural network solution in terms of these techniques
- have an awareness of the computational theory underlying the various methods
- have a working knowledge of one or more neural network simulation packages, and be able to use them to perform a range of computational tasks
- have experience in programming neural network and deep learning applications
- exposure to research techniques in neural networks, deep learning and cognitive science: some topics will be based on research papers and monographs, to which references will be given in the course notes

Textbook

The textbook for this course is:

Deep Learning
By Ian Goodfellow, Yoshua Bengio and Aaron Courville
MIT Press

http://www.deeplearningbook.org (http://www.deeplearningbook.org)
https://mitpress.mit.edu/books/deep-learning (https://mitpress.mit.edu/books/deep-learning)

Assumed Knowledge

The course will assume knowledge of the following mathematical topics:

- Linear Algebra (2.1-2.8)
- Probability (3.1-3.14)
- Calculus and Chain Rule (6.5.2)

Students should study the relevant sections of the textbook (shown in brackets) and, if necessary, try to revise these topics on their own during the first few weeks of the course.

Planned Topics

The planned topics for this course are:

Week	Topic	Textbook
wk 1 Mon	Neuroanatomy, Perceptrons	(1.2, 9.10)
wk 1 Thu	Backpropagation	(4.3, 5.1-5)
wk 2 Mon	Probability and Backprop Variations	(3.1-14, 6.1-5)
wk 2 Thu	PyTorch	-
wk 3 Mon	Hidden Unit Dynamics	(7.11-12, 8.2-3)
wk 3 Thu	Convolutional Networks	(7.9, 9.1-5)
wk 4 Mon	(Labor Day Holiday)	-
wk 4 Thu	Image Processing	(7.4, 8.4, 8.7.1)
wk 5 Mon	Recurrent Networks, LSTM and GRU	(10.2, 10.7, 10.10)
wk 5 Thu	Word Vectors, Language Processing	(10.4, 12.4)
wk 6	(Flexibility Week)	

wk 7 Mon Language Processing
wk 7 Thu Reinforcement Learning (12.5.1.1)
wk 8 Mon Deep Reinforcement Learning (18.1, 20.9)
wk 8 Thu Hopfield Network & Boltzmann Machine (16.7, 17.4, 18.2, 20.1-4)
wk 9 Mon Autoencoders (14.1-5, 20.10.3)
wk 9 Thu Generative Adversarial Networks (20.10.4)
wk10 Mon Extension Topics
wk10 Thu Review

The relevant sections of the textbook are shown in brackets.
The textbook may be supplemented with additional materials for some topics.

Teaching Strategies

Due to social distancing restrictions, the course will be delivered online. Students are required to watch pre-recorded lecture videos before each session. The scheduled class time will take the form of an interactive video chat session, and will be used to briefly summarise the content, deliver additional material, and to answer any questions that you may have about each topic.

The recorded lectures and online sessions introduce you to the various concepts and methods, provide motivating examples to help you understand them, and demonstrate skills and processes. You should not expect to understand the material completely simply by watching the lecture videos. You should also:

- review the lecture material before and after the scheduled class
- discuss the material with fellow students if possible
- read up on the topics covered in each lecture
- complete relevant assignments, exercises and quizzes
- consider exploring the topic on-line by writing and running your own programs
- ask questions in an online consultation session, if you still don't understand the material

Assessment

The assessment for this course will be:

Assignment 1	30%
Assignment 2	30%
Final Exam	40%

The assignments will involve writing code in PyTorch. Please try to install PyTorch on your own laptop, and try to match the environment on the CSE Lab machines as closely as possible:

python3	3.7.3
torch	1.2.0
numpy	1.16.2
sklearn	0.20.2

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:

- Academic Integrity and Plagiarism (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)

Course Evaluation and Development

This course is evaluated each session using the myExperience system. Feedback from previous sessions was generally positive and we have tried to keep the same basic course structure.

The field of Neural Networks and Deep Learning is changing rapidly. This course was substantially redesigned in 2017, and we make an on-going effort to keep the course materials up-to-date and include the latest developments in the field.

All comments and suggestions are welcomed, and will be listened to respectfully and appreciatively.

Enjoy!

Resource created 2 years ago (Monday 31 August 2020, 08:26:48 AM), last modified about a year ago (Tuesday 13 October 2020, 09:54:06 AM).

Comments

(/COMP9444/20T3/forums/search?forum_choice=resource/50779)

(/COMP9444/20T3/forums/resource/50779)

Rittisak Kwanda (/users/z5283054) 2 years ago (Tue Sep 15 2020 15:21:21 GMT+0800 (中国标准时间))

Dear Professor,

What are the release/due dates for both assignments?

Best Regards,

Course Outline

Contents

Course Details

Course Code	COMP6771
Course Title	Advanced C++ Programming
Convenor/Admin	Hayden Smith (/users/z3418003)
Lecturer(s)	Hayden Smith (/users/z3418003) , Matt Stark, Christopher Di Bella
Contact for the course	cs6771@cse.unsw.edu.au
Classes	See here (https://webcms3.cse.unsw.edu.au/COMP6771/20T2/resources/47572)
Help Sessions	See here (https://webcms3.cse.unsw.edu.au/COMP6771/20T2/resources/47572)
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs6771/20T2/ (http://cse.unsw.edu.au/~cs6771/20T2/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6771.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6771.html)

Course Summary

COMP6771 is an advanced programming course teaching practical aspects of intermediate/advanced C++ programming. The course focuses on teaching the fundamentals of C++, followed by exploring powerful abstractions that C++ enables. This course focuses on using abstractions as well as building abstractions.

COMP6771 is focused on modern, practical programming methods and tools. This course is designed for latter year CSE students with a reasonable degree of programming competencies.

The course is heavily supported by Christopher Di Bella (https://www.cjdb.com.au/) , a UNSW CSE graduate who is a well regarded expert on C++. His knowledge and expertise assists in forming and updating the course.

Our aim for students who complete this course satisfactorily is that they are highly competent in understanding C++ and it's core features, being able to build complex programs, data structures, and algorithms with C++, and being ready to immediately move into the workforce in areas that rely heavily on C++.

COMP6771 can be a challenging course for students due to the volume of work to complete in a 10 week period.

Assumed Knowledge

Before commencing this course, students should:

- Be competent in constructing and designing programs in the language C (from COMP1511/1917 or equivalent)
- Be competent in understanding object-oriented (OO) programming methods (from COMP2511)
- Be competent with the basics of git usage (pull, push, add, commit)

We will spend minimum time covering basics of C and OO such as pointers, pointer arithmetic, classes, objects, and memory.

Student Learning Outcomes

After completing this course, students will:

1. Design, build, and test C++ programs
2. Use abstractions (data structures, algorithms) to solve problems efficiently
3. Build abstractions to solve problems efficiently
4. Distinguish good, modern, widely-used practices from more outdated practices
5. Be confident setting up build and testing environments for C++ programs

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

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	Assignment completion and feedback
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Using tutorial and lecture knowledge to complete assignments
Professionals capable of ethical, self- directed practice and independent lifelong learning	Individual assignments completions
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Participation in lectures and group work on assignment

Teaching Strategies

- **Lectures** : 4 hours of online lectures.
- **Tutorials** : 1 hour of tutorial per week to go through examples of work covered in lectures.
- **Assignments:** 3 major assignments that give you an opportunity to practice the lessons.

- **Exam** : Final exam to continue to test theoretical and practical knowledge.

Teaching Technologies

This course uses 4 key pieces of technology to engage you.

- **(Centre) Webcms3** : This is your spring-board, it's the easy place to go back to get to everything you need.
- **(Forum) Piazza** : *All* questions relating to the course should be posted in our Piazza forum (https://piazza.com/class/kakkc32htsm2g2) .
- **(Class) Zoom** : All lectures, consultations, tutorials, and help sessions will be administered via Zoom
- **(Code) Gitlab** : All tutorial and lecture code is hosted on gitlab in your own personal repos, as well as assignments.

Teaching Rationale

Being a latter year CSE course, this course focuses on studying a topic (C++) in depth while assuming that students are mature and independent learners. Because of this a lot of time is focused on content delivery (4 hours of lectures + 1 hour of tutorial), with limited laboratory time. In this way, an expectation is placed on students that they are self-learners and can use the Piazza forum.

The course centres the teaching around 3 programming assignments, worth a total of 50% of the course assessment. We use these assignments to thoroughly put to practice what we teach you. This is a course where failure to make a satisfactory attempt at an assignment will make it difficult to complete further assignments (or the exam) to a high standard.

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	Key Topics	Completion type	Due	Marks
Assignment 1	C++ Basics, STL	**Individual**	Week 3 Friday 8pm	15%
Assignment 2	Operator Overloading, OO, Resource Management	**Individual**	Week 7 Monday 8pm	15%
Assignment 3	Iterators, Exceptions, Templates	**Group (pairs)**	Week 10 Monday 8pm	20%
Final Exam (Hurdle)	All topics	**Individual**	Exam Period	50%

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 adjusted final exam mark is not satisfactory (<50%), the hurdle will be enforced via a grade of UF. If your overall course score is less than 50%, then your grade will be FL.

Content Schedule

Week	Topics

1	Course intro, Getting started
2	C++ Basics, STL
3	Basic Object-Oriented Programming, Operator Overloading
4	Exceptions
5	Resource Management
6	Flexibility week
7	Templates
8	Iterators
9	Advanced Object-Oriented Programming
10	Extension Topics, Exams

Resources for Students

There is no definitive textbook list for this course. Students are able to complete this course without the use of an explicit resource, however, students are encouraged to seek out resources we recommend if they feel they need the help.

If we had to point you to a single resource, it would be:

- *Programming: Principles and Practice Using C++* (https://rads.stackoverflow.com/amzn/click/com/0321992784) (Bjarne Stroustrup, 2nd Edition - May 25, 2014) (**updated for C++11/C++14**) An introduction to programming using C++ by the creator of the language. A good read, that assumes no previous programming experience, but is not only for beginners.

For a more detailed list of resources, you can explore this Stack Overflow (https://stackoverflow.com/questions/388242/the-definitive-c-book-guide-and-list) article. Some of these resources may be better than others, and we do not endorse a particular one.

Special Consideration

If your work in this course is affected by unforeseen adverse circumstances, you should apply for Special Consideration. If your request is reasonable and your work has clearly been impacted, then

- for an assignment, you may be granted an extension
- for the Final Exam, you may be offered a Supplementary Exam

Note the use of the word "may". None of the above is guaranteed. It depends on you making a convincing case that the circumstances have clearly impacted your ability to work.

UNSW handles special consideration requests centrally (in the Student Lifecycle division), so all special consideration requests must be submitted via the UNSW Special Consideration (https://iaro.online.unsw.edu.au/special_consideration/home.login) website.

Special consideration requests must be accompanied by documentation, which will be verified by Student Lifecycle. Do not email the course convenor directly about special consideration.

Extensions on assignments will only be awarded if the majority of the team are affected.

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 24 hours of the exam. If your request is reasonable, then you will be awarded a Supplementary Exam (aka "Supp").

Note that UNSW expects you to be available to sit Supplementary Exams (held in the week Sep 9-13) if required. If you are awarded a Supp and do not attend, then your exam mark will be zero.

For further details on special consideration, see the UNSW Student website (https://student.unsw.edu.au/special-consideration) .

If you are registered with Disability Services, please forward your documentation to Hayden Smith (mailto:mailto:hsmith@cse.unsw.edu.au) within the first two weeks of semester.

Course Evaluation and Development

This course is evaluated each session using the myExperience system. Your feedback will be used to improve future offerings of the course.

Resource created 2 years ago (Sunday 26 April 2020, 10:32:08 PM), last modified 2 years ago (Friday 29 May 2020, 11:38:29 PM).

Comments

(/COMP6771/20T2/forums/search?forum_choice=resource/44758)

(/COMP6771/20T2/forums/resource/44758)

There are no comments yet.