

COMP20050 - Software Engineering Project II

Module Introduction

Ravi Reddy Manumachu
ravi.manumachu@ucd.ie



UCD School of Computer Science.

Scoil na Ríomheolaíochta
UCD.

Outline

- Staff details.
- COMP20500 module descriptor.
- UCD policies.
- Software Engineering Project.



Module Coordinator

Ravi Reddy Manumachu (a.k.a **Ravi**)

Contact: **ravi.manumachu@ucd.ie**

- Assistant Professor, School of Computer Science, UCD.
- More about me below:

<https://people.ucd.ie/ravi.manumachu>



Queries and Troubleshooting

- Please feel free to get in touch for all issues concerning this module in the following order:
 - **Highly recommended to use the classroom lecture for Q&A.**
 - Associate staff (TA and demonstrators).
 - Email.
- If you use email, please include the following:
 - **Your name and student ID.**
 - **Module Code (COMP20050).**



Associate Staff and Support

- Your Teaching Assistant (TA).
 - **Maryam Gillani** <maryam.gillani@ucdconnect.ie>
- Computer Science Support Centre:
<https://csintranet.ucd.ie/CSSC>



COMP20050 Module Descriptor



COMP20050 Definition (1/2)

- The module has two main goals.
- **First goal:**
 - The first goal emphasizes students **working in teams** and applying a modern software engineering methodology called **Scrum** to deliver a working software application with complexity representative of a real-world software project.
 - On completion of the module, the students are expected to understand how professional software developers practice **Scrum**.



COMP20050 Definition (2/2)

- **Second goal:**

- Focuses on evolving a student's knowledge and skills of the core software engineering concepts, principles of **good software architecture, code quality, test quality** and **documentation**.
- On completion of the module, the students are expected to understand **sound software design principles** and apply them in their software applications' design and implementation.



Learning Outcomes (1/2)

- On successful completion of this module, you will be able to:
 - **Design** a **high-level software architecture** of the software application.
 - **Formulate** a **Scrum project plan** divided into **sprints** to develop the software application.



Learning Outcomes (2/2)

- On successful completion of this module, you will be able to:
 - **Create** a fully **functional prototype** of the software application.
 - **Compare** software designs and implementations of the software application.



Programming and Project Management Tools

Programming languages



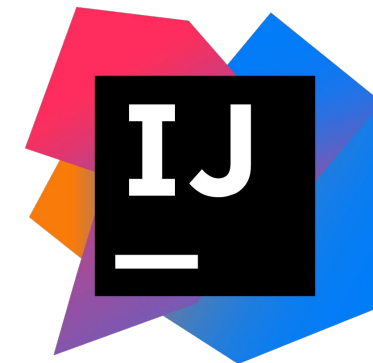
Version Control



Kanban Board



IDE



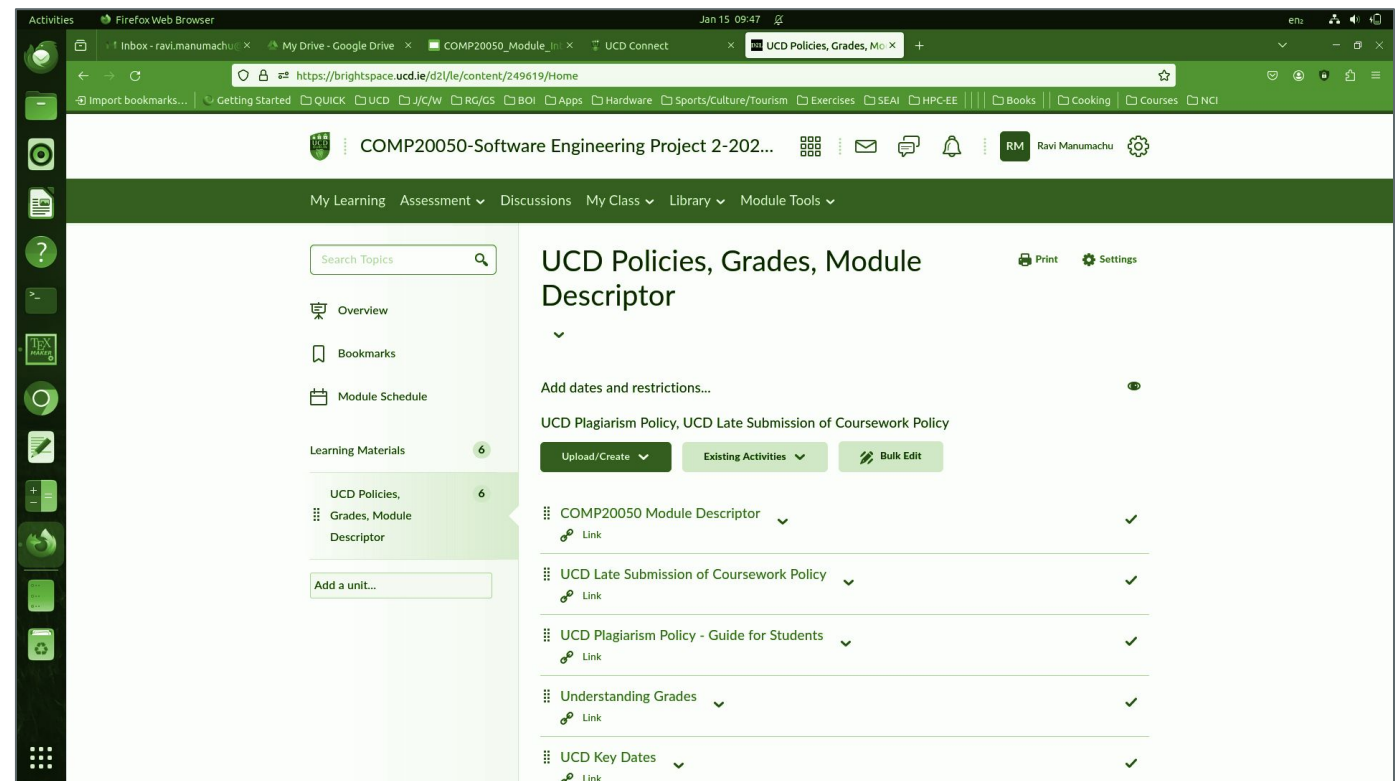
How will I learn? (1/2)

- 24 lectures.
 - Two **1 hour lectures** each week.
- 24 lab sessions.
 - Two **2 hour lab sessions** each week.



How will I learn? (2/2)

- All the learning material will be provided on **Brightspace**.
 - Lecture slides.
 - Additional reading material for each and every lecture.
 - Installation and user manuals for programming tools.
 - Discussion forums.



Am I eligible to take this module?

- **Learning Requirements:**

- Students must be able to program in **Java**. *Students who have no previous experience of Java programming must obtain the permission of the lecturer to attend this module.*

- **Learning Recommendations:**

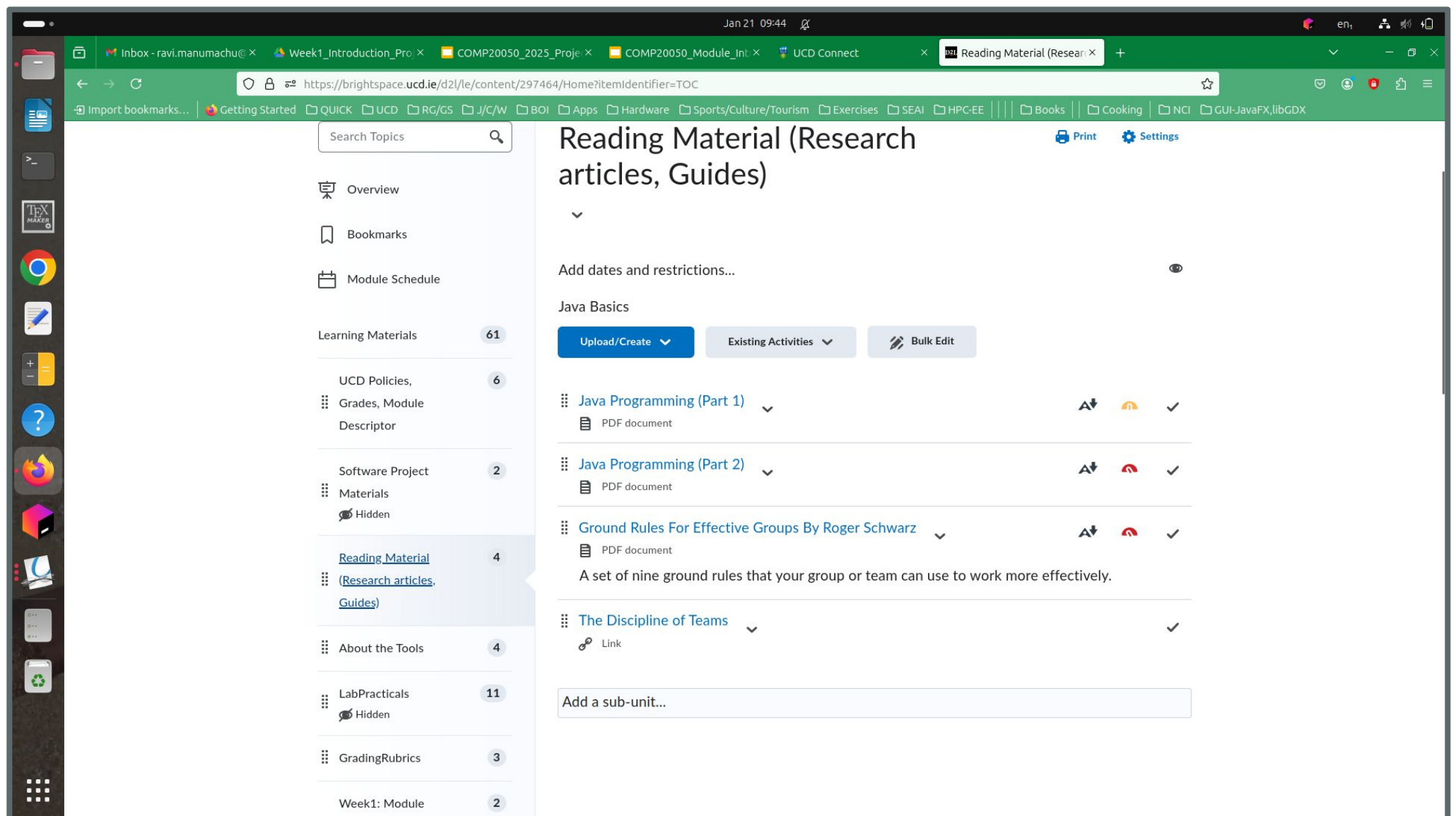
- It is strongly recommended that students take and pass **Programming 1 and 2** and **Software Engineering 1** in the first year before taking this module.
- It is strongly recommended that students take and pass Introduction to **Java** before taking this module.



Java Programming Language

- **Java** is the module's main programming language.
- Polish your Java programming language skills using the reading material below on Brightspace:

<https://brightspace.ucd.ie/d2l/le/content/297464/Home?itemIdentifier=TOC>



The screenshot displays the Brightspace LMS interface. The top navigation bar shows the course title 'COMP20050_2025_Proje' and the current page 'Reading Material (Research articles, Guides)'. The left sidebar contains a list of course items, with 'Reading Material' selected. The main content area shows a list of reading materials, including 'Java Programming (Part 1)', 'Java Programming (Part 2)', 'Ground Rules For Effective Groups By Roger Schwarz', and 'The Discipline of Teams'. Each item has a 'PDF document' icon and a 'Bulk Edit' button. The bottom of the page has an 'Add a sub-unit...' input field.



How will I be assessed? (Group Project 1)

Assessment	Description	Weightage
Group Project 1	Students work in teams employing problem-based Learning approach to research and propose a high-level software architectural design .	15 %
	Based on the high-level design, groups then propose a project plan comprising four sprints .	5 %



Group Project 1: Submission Deadlines

Assessment	Submission	Weightage	Deadline
GP1 High-Level Architecture Design	High-level architecture design (of the software application).	15 %	Week 3
GP1 Project Plan	Project plan comprising product features divided into four sprints.	5 %	Week 3



How will I be assessed? (Group Project 2)

Assessment	Description	Weightage
Group Project 2	<p>Students work in teams of three to develop the software project.</p> <p>The group work is broken into a number of sprint assignments (design and implementation of features).</p>	80 %



Group Project 2: Submission Deadlines

Assessment	Submission	Weightage	Deadline
GP2 Assignment 1	Sprint Assignment 1 (Design and Implementation)	10 %	Week 4
GP2 Assignment 2	Sprint Assignment 2 (Design and Implementation)	10 %	Week 6
GP2 Assignment 3	Sprint Assignment 3 (Design and Implementation)	10 %	Week 8



Group Project 2: Submission Deadlines

Continuous Assessment	Description	Weightage	Deadline
GP2 Assignment 4	Sprint Assignment 4 (Design and Implementation)	10 %	Week 10
GP2 Final Submission	Final Submission	40 %	Week 12



Grading

- Understanding grades.
<https://www.ucd.ie/students/exams/gradingandremediation/understandinggrades/>
- Alternative Linear Conversion Grade Scale 40% Pass.



Alternative Linear Conversion Grade Scale 40% Pass:		
Alternative Linear Conversion Grade Scale 40% Pass (85% = A-)		
Grades	Lower %	Upper %
A+	≥95	100
A	≥90	<95
A-	≥85	<90
B+	≥80	<85
B	≥75	<80
B-	≥70	<75
C+	≥65	<70
C	≥60	<65
C-	≥55	<60
D+	≥50	<55
D	≥45	<50
D-	≥40	<45
E+	≥35	<40
E	≥30	<35
E-	≥25	<30
F+	≥20	<25
F	≥15	<20
F-	≥10	<15
G+	≥5	<10
G	≥0.02	<5
G-	≥0.01	<0.02
NM	0	<0.01
ABS	No work was submitted by the student or the student was absent from assessment	

COMP20050 Indicative Content (1/3)

Weeks	Lecture Description
Week 1	Module Introduction Software Project Description
Week 2	Version Control Software Architectural Design
Week 3	Teamwork and Agile Software Engineering Methodology Scrum Software Engineering Methodology



COMP20050 Indicative Content (2/3)

Weeks	Lecture Description
Week 4-5	JavaFX Framework
Weeks 5-6	LibGDX Framework
Week 7	Clean Testing JUnit Testing Framework



COMP20050 Indicative Content (3/3)

Weeks	Lecture Description
Week 8	Clean Software Architecture Clean Coding Principles: Functions, Names
Week 9	Clean Coding Principles: Classes, Comments
Week 10	Clean Coding Principles: Error Handling
Week 11	S.O.L.I.D Design Principles
Week 12	Component Principles Course Conclusion



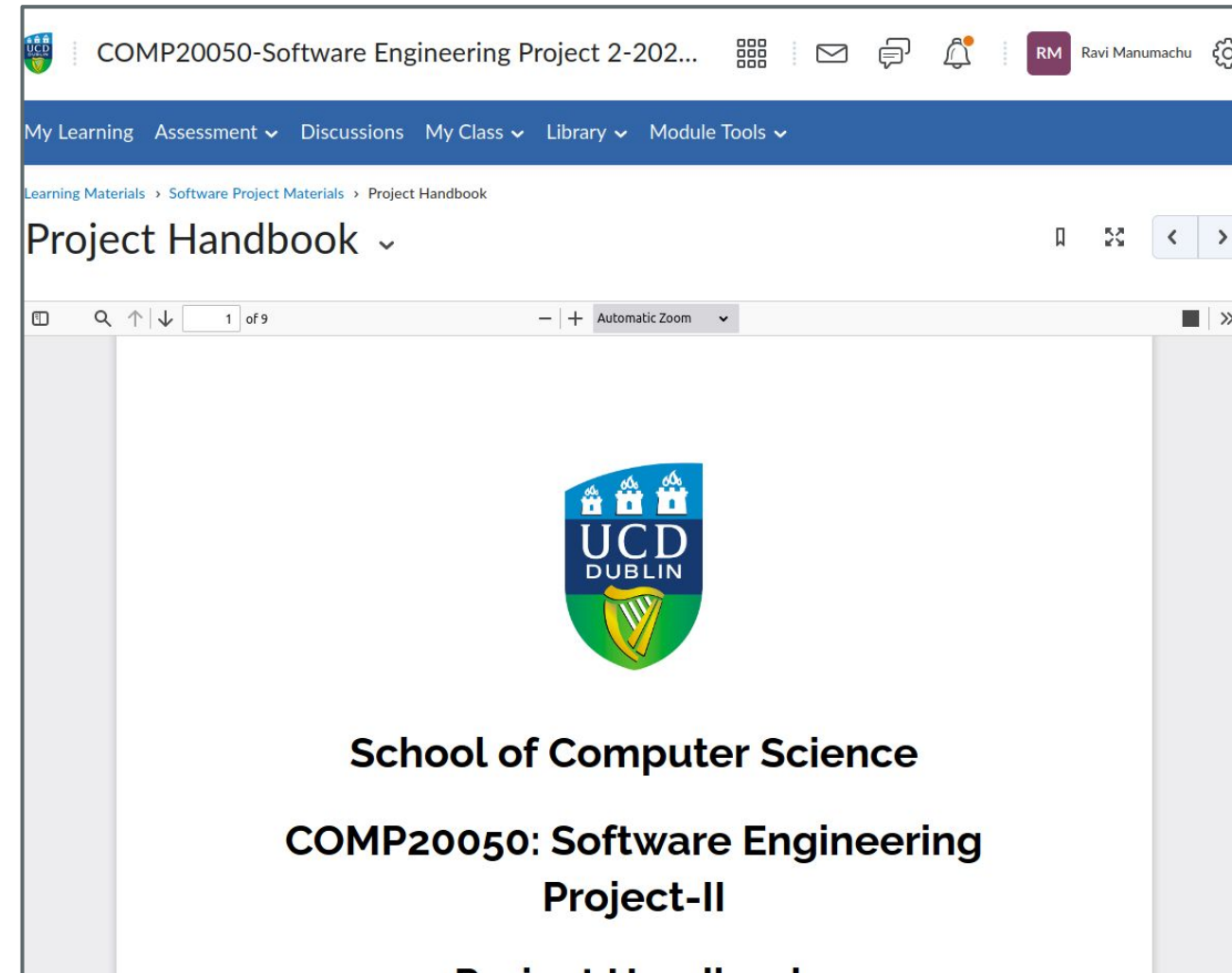
Project Handbook

- All you need to know about the project is provided in this handbook.

My Learning > Software Project Materials > Project Handbook

<https://brightspace.ucd.ie/d2l/le/content/297464/viewContent/3490448/View>

- It is the definitive reference.



Reading Material

Software Engineering

by Ian Sommerville

Publisher: Pearson; 10th edition

Essential Scrum: A Practical Guide to the Most Popular Agile Process

By Kenneth S. Rubin

Publisher: Addison-Wesley Professional; 1st edition

Clean Code: A Handbook of Agile Software Craftsmanship

By Robert C. Martin

Publisher: Prentice Hall; 1st edition

Clean Architecture: A Craftsman's Guide to Software Structure and Design

By Robert C. Martin

Publisher: Addison-Wesley; 1st edition



UCD Policies



UCD Student Code of Conduct

- UCD student code of conduct.

<https://brightspace.ucd.ie/d2l/le/content/249607/viewContent/2596877/View>

- The Student Code of Conduct is established on three principles: **respect**, **responsibility** and **academic integrity**.

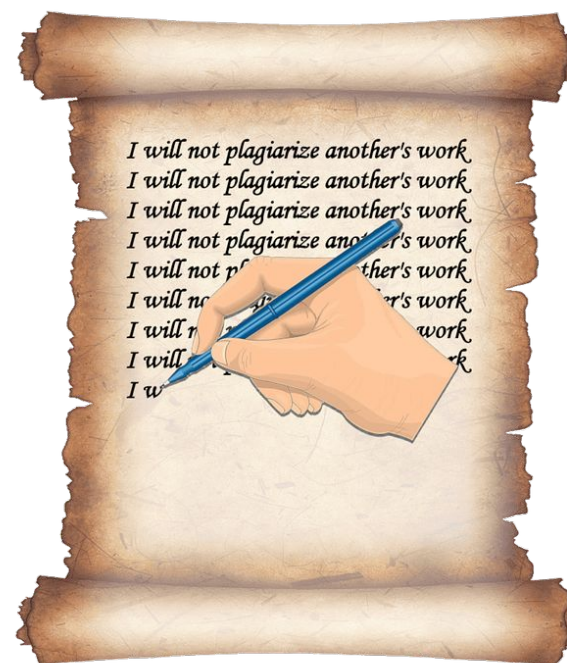


UCD Plagiarism Policy (1/4)

- Plagiarism policy - Student guide.

https://www.ucd.ie/secca/t4media/plagiarism_studentguide.pdf

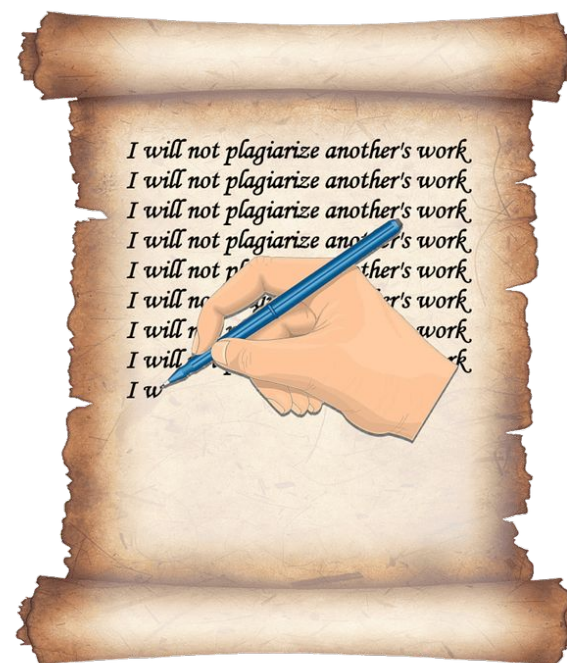
- **Plagiarism is a serious academic offence.**
- It's important that you uphold your own **academic integrity** and that of the University.



UCD Plagiarism Policy (2/4)

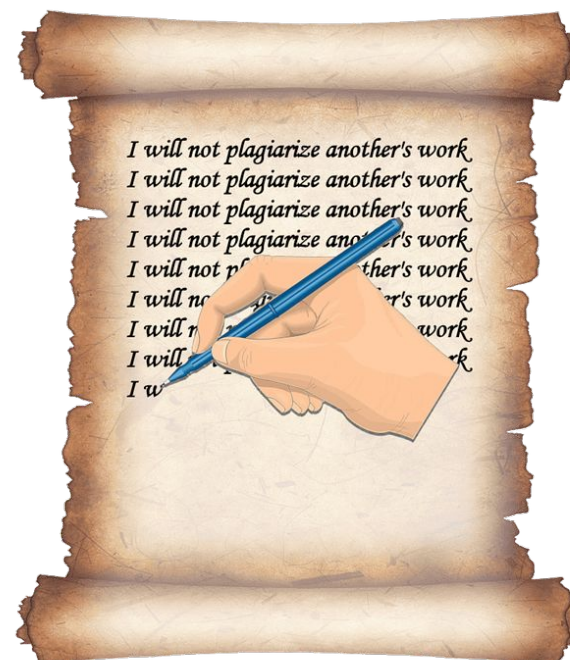
• Examples of plagiarism:

- Failing to cite and acknowledge sources properly.
- Making minor changes to text or paraphrasing from sources like the internet, journals and books, and presenting this as your own words.
- Copying some/all of the work of another student and submitting it as your own work.



UCD Plagiarism Policy (3/4)

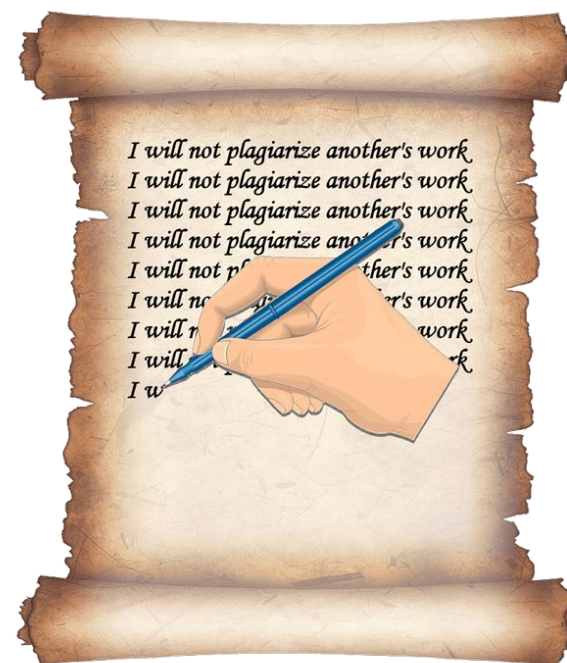
- Our staff and demonstrators are **proactive** in looking for possible plagiarism.
- Suspected plagiarism is investigated by the CS Plagiarism subcommittee.
 - Usually includes an interview with the student(s) involved.
 - **1st offence:** typically **0** or **NM** in the affected components.
 - **2nd offence:** more serious consequences (UCD Disciplinary process).



UCD Plagiarism Policy (4/4)

- **Tips for avoiding plagiarism:**

- **Acknowledge** all sources.
- **Paraphrase correctly**; express the information of others in your own words.
- **Quote correctly**; when directly quoting from a text include appropriate quotation marks or indentation.



UCD Teaching and Learning and ChatGPT

- Please read the following guide on **Generative Artificial Intelligence in Learning and Assessment**.
https://www.ucd.ie/teaching/t4media/Generative_Artificial_Intelligence_Quick_Guide.pdf
 - Designing in: Using and Acknowledging AI to enhance Student Learning.
 - **Designing to Minimise: Discouraging the use of AI in Student Learning.**
- **Any essay/assignment submitted using generative AI tools will not be accepted.**



UCD Late Submission of Coursework Policy

- Please read the following UCD late submission of coursework policy document.

<https://brightspace.ucd.ie/d2l/le/content/249607/viewContent/2586204/View>

- **No extensions** to deadlines for any assignments/submissions.
- Please read **Extenuating Circumstances** policy if it applies to your late submissions.

https://hub.ucd.ie/usis/!W_HU_MENU.P_PUBLISH?p_tag=GD-DOCLAND&ID=126



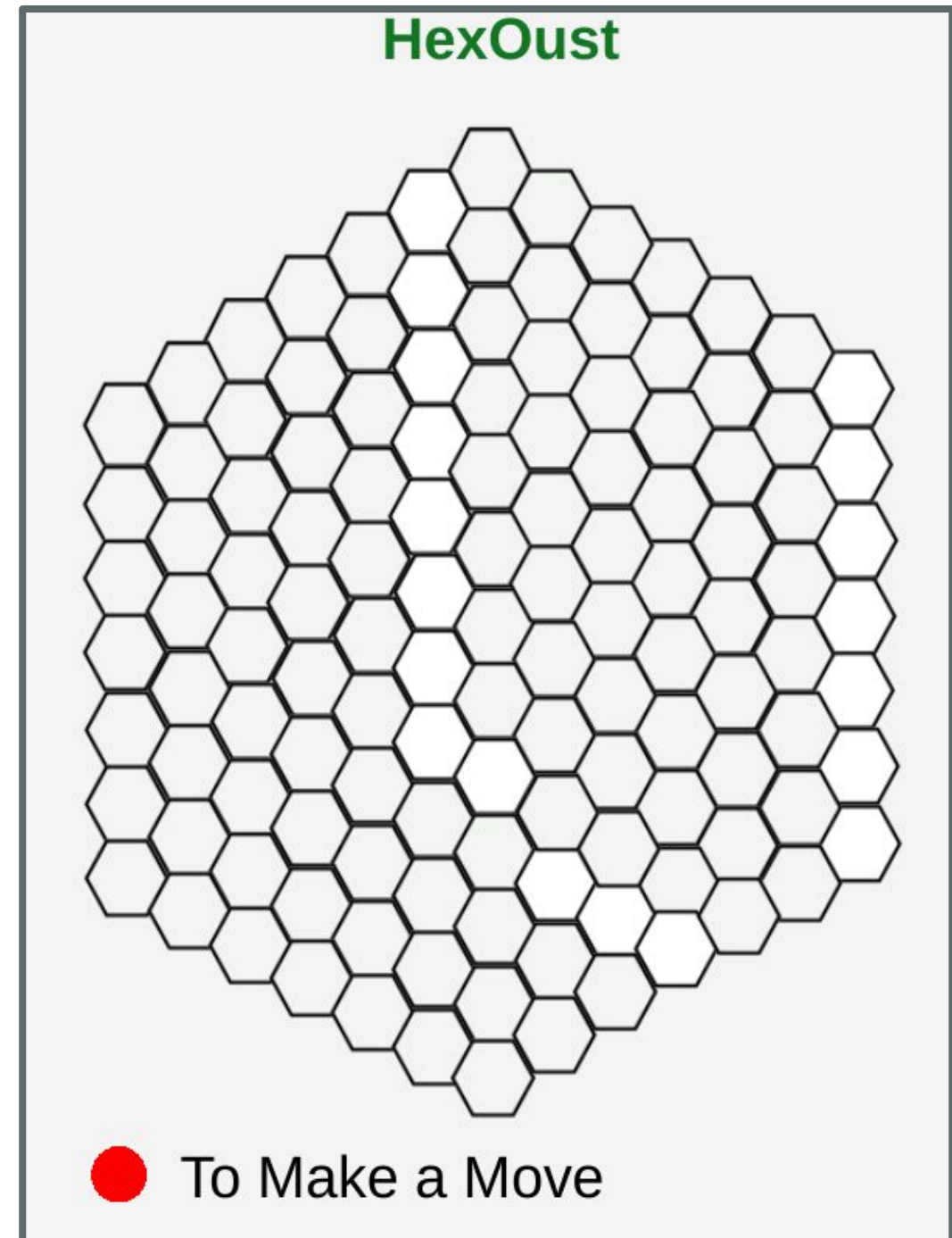
Software Engineering Project

Creating Groups and GitHub



Software Engineering Project

- You will be developing a software implementation of **Hex Oust** in **Java Programming Language**.
<https://mindsports.nl/index.php/the-pit/614-hexoust>
- Detailed description of **Hex Oust** to follow in the next lecture.



Form Your Group

- You will execute this project in a **group** of **three** members.
- Self-select your group at **My Class > My Groups** on **Brightspace**.
- Send your group details to our TA (user name: **gillanimaryam**).
- After the group selection expiry date (next week), any class members who have not self-selected will be automatically and randomly allocated to a group.



Project Source Version Control (1/2)

- **Git** and **GitHub** will be the source code repo and version control for this project.
- Setup **Git** on your computer. Setup your **GitHub account** online.
- Setup a GitHub repo for the project. **THIS REPO MUST BE PRIVATE.**
- The repo name should include your Group numbers allocated on Brightspace.



Project Source Version Control (2/2)

- Share the repo with your team members.
- Give repo access to our TA (user name: **gillanimaryam**) and the Module Coordinator (user name: **ravimanumachu**).
- GitHub must be used for source code control for the duration of the project.
- **GitHub not used will result in 2 grade point deduction.**



Q&A



To follow...

Software Engineering Project Description

