Bachelor of Science in Computer Science

Handbook

April 2026

Introduction

Bachelor of Science in Computer Science

This degree is designed for undergraduate students considering a career in technology or the innovation economy. The programme is suited to those with little or no background in computer programming, requiring only EQF 4 level mathematical knowledge. It prepares students for substantial academic study at EQF 5–6 level, leading to a Bachelor of Science in Computer Science. Students may pursue the general programme or choose a specialisation in **Human-Computer Interaction (HCI)** — focusing on user-centred digital design, interaction design, and usability — or in **Cyber Security** — covering network security, cryptography, defensive tactics, and secure systems development.

Entry requirements

Education Requirements

Candidates must have an EQF 4 level education; or, in cases of exceptional talent (supported by a letter of recommendation and/or portfolio of evidence), an EQF 3 level education with a clear prospect of successful engagement at EQF 5–6 level; or an EQF 3 level education and at least 3 years of relevant industry experience. Students with relevant experience or alternative forms of learning who hold less than EQF 4 education may apply for Recognition of Prior Learning (RPL) at the time of admission.

Language Requirements

The programme may be taught in English, Spanish, or Ukrainian. Whatever the language of instruction, students must demonstrate equivalent linguistic competency at an IELTS 6 equivalent (e.g. DELE or SIELE for Spanish).

Instructional design

Teaching: Teaching is delivered through a blend of synchronous and asynchronous engagement. Synchronous components include live sessions with an instructor, supervised activities, and real-time communication. Asynchronous components include scholarly articles and study materials in the VLE, recorded lectures, and communication via chat forums and discussion boards. This mixed approach provides flexibility for students from diverse work-life situations while maintaining motivation through regular synchronous contact.

Assessment: Assessment combines formative and summative elements throughout each module. Formative assessment consists of weekly assignments. Summative assessment comprises final assignments: one for 3 ECTS modules, two for 6 ECTS modules, and one for 12 ECTS practical/project-based modules.

Degree structure

The degree consists of 180 ECTS distributed across three years of full-time study (or six years part-time). Modules are delivered across EQF levels 5 and 6, progressing from foundational to advanced topics. Students select a general track or one of two specialisations (HCI or Cyber Security).

Module	ECTS	Level
Introduction to Programming in Python	3	EQF 5
Optimizing Your Learning	3	EQF 5
Web Foundations	3	EQF 5
Mathematical Thinking	6	EQF 5
Collaborating for Impact	3	EQF 5
Communicating for Success	3	EQF 5
Speaking	3	EQF 5
Open Source Software	3	EQF 5
Software Engineering	6	EQF 5
Writing	6	EQF 5
Web Development Fundamentals	6	EQF 5
Mobile Development 1	6	EQF 5
Introduction to Cyber Security	6	EQF 5
Web Application Development	6	EQF 5
Front End Web Development	6	EQF 5

Product Management and Design	6	EQF 5
Team Software Project	6	EQF 5
Data Structures and Algorithms 1	6	EQF 5
Engineering for Development	6	EQF 6
Discrete Math	6	EQF 6
Computer Systems	6	EQF 6
Challenge Studio 1	6	EQF 6
Introduction to Data Science	6	EQF 6
Programming in Python	6	EQF 6
Network and Computer Security	6	EQF 6
Challenge Studio 2	6	EQF 6
Data Engineering	6	EQF 6
Fundamental Cyber Attacks and Defensive Tactics	6	EQF 6
Backend Development	6	EQF 6
iOS App Development	6	EQF 6
Ethics for Tech	6	EQF 6
Android App Development	6	EQF 6
Artificial Intelligence	6	EQF 6

Machine Learning	6	EQF 6
Interaction Design	6	EQF 6
Designing Your Future	3	EQF 6
Engineering Your Career	3	EQF 6
Mobile Development 2	6	EQF 6
Advanced Cyber Attacks and Defensive Tactics	6	EQF 6
Data Structures and Algorithms 2	6	EQF 6

Module Descriptions

1. Introduction to Programming in Python

This course is intended for students with little or no programming experience. It aims to help students develop an appreciation for programming as a problem-solving tool and to provide a foundation in Python programming. Students learn to think algorithmically, solve problems efficiently, and prepare for further computer science studies.

Learning Outcomes

1. Ability to write basic Python programs to solve simple computational problems
2. Understanding of fundamental programming concepts including variables, data types, control flow, and functions
3. Problem-solving skills using algorithmic thinking
4. Ability to debug and test simple Python programs

2. Optimizing Your Learning

This course is intended to transform incoming first-year students into effective and empowered self-directed learners. It develops metacognitive skills, time management, and learning strategies essential for success in higher education.

Learning Outcomes

1. Ability to plan and manage personal learning effectively
2. Metacognitive awareness of own learning processes
3. Effective use of study resources and university support services

4. Self-assessment and reflection on learning progress

3. Web Foundations

An introduction to web technologies including HTML, CSS, and basic JavaScript. Students learn the fundamentals of how web applications work, understand client-server architecture, and create static and simple interactive web pages.

Learning Outcomes

1. Ability to create well-structured HTML documents
2. Understanding of CSS for styling and layout
3. Basic JavaScript interactivity
4. Understanding of web architecture and HTTP protocols

4. Mathematical Thinking

Covers discrete mathematics concepts essential for computer science, including logic, set theory, graph theory, and combinatorics. Develops mathematical reasoning and proof techniques.

Learning Outcomes

1. Understanding of mathematical logic and formal reasoning
2. Ability to apply graph theory concepts to computational problems
3. Understanding of combinatorics and probability fundamentals
4. Ability to construct and verify mathematical proofs

5. Collaborating for Impact

Develops teamwork and collaboration skills essential for professional software development. Covers team dynamics, project management basics, and effective communication within teams.

Learning Outcomes

1. Ability to work effectively in diverse teams
2. Understanding of project management fundamentals
3. Conflict resolution and negotiation skills
4. Ability to contribute meaningfully to team objectives

6. Communicating for Success

Develops written and oral communication skills for technical and professional contexts, including technical writing, presentations, and professional correspondence.

Learning Outcomes

1. Ability to write clear technical documentation
2. Proficiency in delivering technical presentations
3. Professional written communication skills
4. Ability to adapt communication for different audiences

7. Speaking

Focuses on developing public speaking and presentation skills, including confidence, clarity, and engaging delivery techniques for technical and general audiences.

Learning Outcomes

1. Ability to deliver clear, structured presentations
2. Confidence in public speaking
3. Effective use of presentation aids and visual materials
4. Ability to handle questions and audience interaction

8. Open Source Software

Introduction to open source software philosophy, development practices, and contribution. Students learn about version control systems, collaborative development, and open source licensing.

Learning Outcomes

1. Understanding of open source principles and licensing
2. Ability to use version control systems including Git
3. Ability to contribute to open source projects
4. Understanding of open source community practices

9. Software Engineering

Covers the software development lifecycle, design patterns, code quality, and engineering practices. Introduces students to systematic approaches to software development beyond basic programming.

Learning Outcomes

1. Understanding of software development lifecycle methodologies
2. Ability to apply design patterns in code
3. Code quality and refactoring practices
4. Ability to document software designs and specifications

10. Writing

Develops academic and professional writing skills including essays, reports, technical documentation, and creative writing. Covers grammar, structure, and effective written communication.

Learning Outcomes

1. Ability to write well-structured academic essays and reports
2. Proficiency in technical writing and documentation
3. Understanding of different writing styles and genres
4. Editing and revision skills for clarity and effectiveness

11. Web Development Fundamentals

Builds on web foundations with practical development of interactive web applications using modern frameworks and techniques. Covers responsive design, client-side programming, and web best practices.

Learning Outcomes

1. Ability to develop responsive, user-friendly web applications
2. Understanding of modern web development frameworks
3. Proficiency in client-side JavaScript programming
4. Understanding of web accessibility and performance optimisation

12. Mobile Development 1

Introduction to mobile application development. Covers mobile platforms, app lifecycle, user interface design for mobile, and development of basic mobile applications.

Learning Outcomes

1. Understanding of mobile platform architecture and constraints
2. Ability to design user interfaces for mobile devices
3. Proficiency in developing basic mobile applications
4. Understanding of mobile development tools and frameworks

13. Introduction to Cyber Security

Foundational concepts in cybersecurity including basic cryptography, network security, common threats, and security best practices. Introduces students to a security mindset and defensive thinking.

Learning Outcomes

1. Understanding of basic cryptographic concepts and applications
2. Awareness of common security vulnerabilities and threats
3. Knowledge of security best practices and policies
4. Ability to identify basic security risks in systems

14. Web Application Development

Advanced web development covering backend frameworks, databases, and full-stack development. Students build complete web applications with server-side logic and data persistence.

Learning Outcomes

1. Proficiency in backend web framework development
2. Understanding of relational and NoSQL databases
3. Ability to design and implement RESTful APIs
4. Full-stack application development capability

15. Front End Web Development

Specialised study of client-side web development technologies. Covers modern JavaScript frameworks, component-based architecture, state management, and advanced front-end techniques.

Learning Outcomes

1. Advanced proficiency in JavaScript and front-end frameworks
2. Ability to build complex interactive user interfaces
3. Understanding of front-end architecture and design patterns
4. Proficiency in front-end tooling and build systems

16. Product Management and Design

Combines product management and user experience design concepts. Covers product strategy, user research, wireframing, prototyping, and iterative design processes.

Learning Outcomes

1. Ability to conduct user research and create user personas
2. Proficiency in creating wireframes and prototypes
3. Understanding of design thinking and iterative design
4. Ability to translate user needs into product requirements

17. Team Software Project

Capstone project requiring students to work in teams to develop a substantial software application. Integrates learning from previous modules and emphasises project management, teamwork, and delivery.

Learning Outcomes

1. Ability to manage and execute complex software projects
2. Effective team collaboration and communication
3. Project planning and risk management
4. Ability to integrate multiple technologies in a cohesive solution

18. Data Structures and Algorithms 1

Fundamental data structures and algorithms including arrays, linked lists, stacks, queues, sorting, and searching. Emphasises understanding of algorithmic complexity and efficient implementations.

Learning Outcomes

1. Proficiency in implementing fundamental data structures
2. Understanding of algorithm analysis and Big-O notation
3. Ability to select appropriate data structures for problems
4. Proficiency in sorting and searching algorithms

19. Engineering for Development

Focuses on applying engineering principles and technologies to solve real-world problems in developing contexts. Covers sustainable development, appropriate technology, and the social impact of software.

Learning Outcomes

1. Ability to assess technical feasibility for development projects
2. Understanding of sustainable and appropriate technology principles
3. Awareness of social and ethical implications of technology
4. Ability to design solutions for resource-constrained environments

20. Discrete Math

Advanced discrete mathematics including advanced graph theory, formal languages, algorithm analysis, and computational complexity. Provides the theoretical foundation for computer science.

Learning Outcomes

1. Advanced understanding of graph algorithms and applications
2. Ability to analyse algorithm complexity and efficiency
3. Understanding of formal language theory basics
4. Ability to apply discrete mathematics to algorithmic problems

21. Computer Systems

Covers computer architecture, operating systems fundamentals, memory management, process scheduling, and system-level programming. Provides an understanding of how computers execute programs at a low level.

Learning Outcomes

1. Understanding of computer architecture and processor design
2. Knowledge of operating system concepts and functionality
3. Proficiency in system-level programming
4. Ability to optimise code for system performance

22. Challenge Studio 1

Project-based learning module in which students tackle real-world computational challenges. Emphasises creative problem-solving, interdisciplinary collaboration, and innovation.

Learning Outcomes

1. Ability to approach unfamiliar problems systematically
2. Creative problem-solving and innovation skills
3. Ability to collaborate across disciplines
4. Ability to present and defend solutions

23. Introduction to Data Science

Fundamentals of data analysis, statistics, and data visualisation. Covers data collection, cleaning, analysis techniques, and interpretation of results for informed decision-making.

Learning Outcomes

1. Proficiency in statistical analysis and hypothesis testing

2. Ability to create meaningful data visualisations
3. Understanding of data quality and cleaning techniques
4. Ability to draw valid conclusions from data analysis

24. Programming in Python

Advanced Python programming covering functional programming, object-oriented design, testing, and practical software development practices. Builds on foundational programming concepts.

Learning Outcomes

1. Advanced proficiency in Python programming
2. Understanding of functional and object-oriented paradigms
3. Proficiency in test-driven development and code quality
4. Ability to use Python libraries and frameworks effectively

25. Network and Computer Security

Advanced cybersecurity covering network protocols, intrusion detection, firewall configuration, penetration testing basics, and security architecture design.

Learning Outcomes

1. Understanding of network security protocols and architecture
2. Ability to configure and manage security systems
3. Proficiency in security testing and vulnerability assessment
4. Understanding of security incident response

26. Challenge Studio 2

Advanced project-based learning building on Challenge Studio 1. Students tackle more complex challenges requiring integration of multiple technical and non-technical competencies.

Learning Outcomes

1. Advanced problem-solving across complex domains
2. Ability to lead and manage project teams
3. Strategic thinking and long-term planning
4. Ability to present sophisticated solutions to stakeholders

27. Data Engineering

Large-scale data processing and pipeline development. Covers distributed computing, data warehousing, ETL processes, and tools for handling big data.

Learning Outcomes

1. Understanding of distributed computing principles
2. Proficiency in ETL development and data pipelines
3. Ability to design scalable data architectures

4. Experience with big data technologies and platforms

28. Fundamental Cyber Attacks and Defensive Tactics

Detailed study of common cyberattacks including malware, SQL injection, social engineering, and DDoS. Covers corresponding defensive strategies and mitigation techniques.

Learning Outcomes

1. Deep understanding of attack vectors and methodologies
2. Ability to develop and implement defensive strategies
3. Proficiency in security assessment and vulnerability identification
4. Understanding of incident response and remediation

29. Backend Development

Advanced server-side development covering scalable architectures, database optimisation, API design, and deployment practices for production systems.

Learning Outcomes

1. Proficiency in building scalable backend systems
2. Advanced database design and optimisation
3. Understanding of microservices and distributed architectures
4. Ability to deploy and maintain production systems

30. iOS App Development

Development of native iOS applications using Swift and iOS frameworks. Covers app architecture, UI design for iOS, and publishing to the App Store.

Learning Outcomes

1. Proficiency in the Swift programming language
2. Ability to design and develop native iOS applications
3. Understanding of iOS frameworks and development patterns
4. Ability to publish and manage apps in the App Store

31. Ethics for Tech

Examines the ethical implications of technology including privacy, algorithmic bias, data ethics, and societal impact. Develops ethical reasoning in technology decision-making.

Learning Outcomes

1. Understanding of privacy and data ethics principles
2. Ability to identify and address algorithmic bias
3. Understanding of technology's societal and environmental impact
4. Ability to make ethical decisions in technology development

32. Android App Development

Development of native Android applications using Java or Kotlin. Covers the Android framework, app lifecycle, UI development, and publishing to the Google Play Store.

Learning Outcomes

1. Proficiency in Java or Kotlin programming for Android
2. Ability to design and develop native Android applications
3. Understanding of the Android framework and development patterns
4. Ability to publish and manage apps in the Google Play Store

33. Artificial Intelligence

Foundational concepts in artificial intelligence including search algorithms, knowledge representation, reasoning systems, and introductory machine learning applications.

Learning Outcomes

1. Understanding of AI algorithms and problem-solving approaches
2. Ability to design knowledge representation systems
3. Introduction to machine learning concepts and applications
4. Ability to evaluate and implement AI solutions

34. Machine Learning

Advanced study of machine learning algorithms, model training, validation, and deployment. Covers supervised and unsupervised learning, neural networks, and practical ML applications.

Learning Outcomes

1. Proficiency in machine learning algorithms and libraries
2. Ability to train and validate ML models
3. Understanding of neural networks and deep learning basics
4. Ability to deploy ML models in production systems

35. Interaction Design

Specialised study in designing digital interactions and user experiences. Covers interaction patterns, usability principles, and evaluation methods for digital products.

Learning Outcomes

1. Advanced proficiency in UX/UI design principles
2. Ability to conduct user research and testing
3. Proficiency in design tools and prototyping software
4. Ability to evaluate and improve user interactions

36. Designing Your Future

Career planning and professional development module. Helps students identify career goals, develop personal brands, and prepare for transition to employment or further study.

Learning Outcomes

1. Ability to identify and articulate career goals
2. Understanding of professional development pathways
3. Ability to create professional portfolios and CVs
4. Networking and job-search competencies

37. Engineering Your Career

Focused career preparation for engineering and technology roles. Covers technical interviewing, project portfolio development, and positioning for technology industry positions.

Learning Outcomes

1. Ability to demonstrate technical competencies effectively
2. Proficiency in technical interview skills
3. Ability to build and present technology portfolios
4. Understanding of career progression in the tech industry

38. Mobile Development 2

Advanced mobile development covering cross-platform frameworks, performance optimisation, advanced UI patterns, and publication strategies for multiple platforms.

Learning Outcomes

1. Proficiency in cross-platform mobile development frameworks
2. Advanced UI/UX design for mobile
3. Ability to optimise mobile application performance
4. Advanced skills in app publication and maintenance

39. Advanced Cyber Attacks and Defensive Tactics

Advanced cybersecurity covering sophisticated attacks, advanced persistence, forensics, and defence-in-depth strategies. Prepares students for advanced security roles.

Learning Outcomes

1. Understanding of advanced attack methodologies
2. Ability to analyse and respond to sophisticated threats
3. Proficiency in digital forensics and incident investigation
4. Ability to architect comprehensive security solutions

40. Data Structures and Algorithms 2

Advanced data structures and algorithms including trees, graphs, dynamic programming, and advanced algorithmic techniques. Builds on foundational concepts for complex problem-solving.

Learning Outcomes

1. Proficiency in advanced data structures including trees and graphs
2. Advanced algorithmic problem-solving skills
3. Understanding of dynamic programming and optimisation techniques
4. Ability to analyse and implement complex algorithms

Internships policy

Internships must be a genuine extension of the student's academic programme, providing opportunity to apply theoretical knowledge to substantive projects directly related to their field of study. Internships consisting primarily of administrative or routine tasks will not be approved.

Every internship must have a defined start date, end date, and formal learning plan with objectives agreed in advance by the student, the host organisation, and the relevant college. Responsibilities and task complexity should increase over time. Each student must be assigned a named supervisor within the host organisation who holds relevant expertise and is responsible for providing regular guidance and feedback.

Woolf prioritises paid internships to ensure equitable access regardless of socioeconomic background. Unpaid internships may only be approved where they constitute a genuine learning opportunity and do not displace the work of a paid employee.

Programmatic standards

Day-to-day management sits with the relevant college. Each college must have a designated Woolf contact responsible for vetting and approving all host organisations and placements before any internship may proceed. Colleges are responsible for matching students to approved positions.

Students must complete pre-internship preparation before commencing a placement, which may include CV writing, interview support, and other instruction as necessary. Virtual internships are encouraged to widen access beyond geographical constraints; support systems must address the challenges of remote work, including cross-timezone communication and fostering professional belonging.

Programme effectiveness must be evaluated on an ongoing basis. Formal evaluations will be collected from students, host supervisors, and academic advisors, and will inform curriculum design and programme improvement.

Grading Scheme

General Marking Criteria and Classification

Marking of student work keeps in view the scale of work that the student can reasonably be expected to have undertaken in order to complete the task.

The assessment of work for the course is defined according to the following rubric of general criteria:

1. **Engagement:**
    - Directness of engagement with the question or task
    - Range of issues addressed or problems solved
    - Depth, complexity, and sophistication of comprehension of issues and implications of the questions or task

    - Effective and appropriate use of imagination and intellectual curiosity
2. **Argument or solution:**
    - Coherence, mastery, control, and independence of work
    - Conceptual and analytical precision
    - Flexibility, i.e., discussion of a variety of views, ability to navigate through challenges in creative ways
    - Completion leading to a conclusion or outcome
    - Performance and success of the solution, where relevant
3. **Evidence (as relevant):**
    - Depth, precision, detail, range and relevance of evidence cited
    - Accuracy of facts
    - Knowledge of first principles and demonstrated ability to reason from them
    - Understanding of theoretical principles and/or historical debate
    - Critical engagement with primary and/or secondary sources
4. **Organisation & Presentation:**
    - Clarity and coherence of structure
    - Clarity and fluency of writing, code, prose, or presentation (as relevant)
    - Correctness of conformity to conventions (code, grammar, spelling, punctuation, or similar relevant conventions)

Definition of marks

97-100

Work will be so outstanding that it could not be better within the scope of the assignment. These grades will be used for work that shows exceptional excellence in the relevant domain; including (as relevant): remarkable sophistication and mastery, originality or creativity, persuasive and well-grounded new methods or ideas, or making unexpected connections or solutions to problems.

94-96

Work will excel against each of the General Criteria. In at least one area, the work will be merely highly competent.

90-93

Work will excel in more than one area, and be at least highly competent in other respects. It must be excellent and contain: a combination of sophisticated engagement with the issues; analytical precision and independence of solution; go beyond paraphrasing or boilerplate code techniques; demonstrating quality of awareness and analysis of both first principles or primary evidence and scholarly debate or practical tradeoffs; and clarity and coherence of presentation. Truly outstanding work measured against some of these criteria may compensate for mere high competence against others.

87-89

Work will be at least very highly competent across the board, and excel in at least one group of the General Criteria. Relative weaknesses in some areas may be compensated by conspicuous strengths in others.

84-86

Work will demonstrate considerable competence across the General Criteria. They must exhibit some essential features of addressing the issue directly and relevantly across a good range of aspects; offer a coherent solution or argument involving (where relevant) consideration of alternative approaches; be substantiated with accurate use of resources (including if relevant, primary evidence) and contextualisation in debate (if relevant); and be clearly presented. Nevertheless, additional strengths (for instance, the range of problems addressed, the sophistication of the arguments or solutions, or the use of first principles) may compensate for other weaknesses.

80-83

Work will be competent and should manifest the essential features described above, in that they must offer direct, coherent, substantiated and clear arguments; but they will do so with less range, depth, precision and perhaps clarity. Again, qualities of a higher order may compensate for some weaknesses.

77-79

Work will show solid competence in solving problems or providing analysis. But it will be marred by weakness under one or more criteria: failure to fully solve the problem or discuss the question directly; some irrelevant use of technologies or citing of information; factual error, or error in selection of technologies; narrowness in the scope of solution or range of issues addressed or evidence adduced; shortage of detailed evidence or engagement with the problem; technical performance issues (but not so much as to prevent operation); poor organisation or presentation, including incorrect conformity to convention or written formatting.

74-76

Work will show evidence of some competence in solving problems or providing analysis. It will also be clearly marred by weakness in multiple General Criteria, including: failure to solve the problem or discuss the question directly; irrelevant use of technologies or citing of information; factual errors or multiple errors in selection of technologies; narrowness in the scope of solution or range of issues addressed or evidence adduced; shortage of detailed evidence or engagement with the problem; significant technical performance issues (but not so much as to prevent operation); poor organisation or presentation, including incorrect conformity to convention or written formatting. They may be characterised by unsubstantiated assertion rather than argument, or by unresolved contradictions in the argument or solution.

70-73

Work will show evidence of competence in solving problems or providing analysis, but this evidence will be limited. It will be clearly marred by weakness in multiple General Criteria. It will still make substantive progress in addressing the primary task or question, but the work will lack a full solution or directly address the task; the work will contain irrelevant material; the work will show multiple errors of fact or judgment; and the work may fail to conform to conventions.

67-69

Work will fall down on a number of criteria, but will exhibit some of the qualities required, such as the ability to grasp the purpose of the assignment, to deploy substantive information or solutions in an effort to complete the assignment; or to offer some coherent analysis or work towards the assignment. Such qualities will not be displayed at a high level, and may be marred by irrelevance, incoherence, major technical performance issues, error and poor organisation and presentation.

64-66

Work will fall down on a multiple General Criteria, but will exhibit some vestiges of the qualities required, such as the ability to see the point of the question, to deploy information, or to offer some coherent work. Such qualities will be substantially marred by irrelevance, incoherence, error and poor organisation and presentation.

60-63

Work will display a modicum of knowledge or understanding of some points, but will display almost none of the higher qualities described in the criteria. They will be marred by high levels of factual or technology error and irrelevance, generalisation or boilerplate code and lack of information, and poor organisation and presentation.

0-60

Work will fail to exhibit any of the required qualities. Candidates who fail to observe rubrics and rules beyond what the grading schemes allow for may also be failed.

Indicative equivalence table

US GPA	US Grade	US Percent	UK Mark	UK UG Classification	UK PG Classification	Malta Grade	Malta Mark	Malta Classification	Swiss Grade
4	A+	97 - 100	70+	First	Distinction	A	80-100%	First class honours	6.0
3.9	A	94-96				B	70-79%	Upper-second class honours	
3.7	A-	90-93							5.5
3.3	B+	87-89	65-69	Upper Second	Merit	C	55-69%	Lower-second class honours	
3	B	84-86	60-64						
2.7	B-	80-83	55-59	Lower Second	Pass				5
2.3	C+	77-79	50-54			D	50-54%	Third-class honours	
2	C	74–76	45-49	Third	Pass				
1.7	C-	70–73	40-44						
1.3	D+	67–69	39-	Fail	Fail				
1	D	64–66							
0.7	D-	60–63							
0	F	Below 60				F			

Synchronous Adjustments Template

Synch discussions may affect the mark on submitted assignments: written work is submitted in advance, and a discussion follows. This provides students an opportunity to clarify and explain their written claims, and it also tests whether the work is a product of the student’s own research or has been plagiarised.

The synchronous discussion acts to shift the recorded mark on the submitted assignment according to the following rubric:

+3

Up to three points are added for excellent performance; the student displays a high degree of competence across a range of questions, and excels in at least one group of criteria. Relative weaknesses in some areas may be compensated by conspicuous strengths in others.

+/- 0

The marked assignment is unchanged for fair performance. Answers to questions must show evidence of some solid competence in expounding evidence and analysis. But they will be marred by weakness under one or more criteria: failure to discuss the question directly; appeal to irrelevant information; factual error; narrowness in the range of issues addressed or evidence adduced; shortage of detailed evidence; or poor organisation and presentation, including consistently incorrect grammar. Answers may be characterised by unsubstantiated assertion rather than argument, or by unresolved contradictions in the argument.

- 3 (up to three points)

Up to three are subtracted points for an inability to answer multiple basic questions about themes in the written work. Answers to questions will fall down on a number of criteria, but will exhibit some vestiges of the qualities required, such as the ability to see the point of the question, to deploy information, or to offer some coherent analysis towards an argument. Such qualities will not be displayed at a high level or consistently, and will be marred by irrelevance, incoherence, error and poor organisation and presentation.

0 (fail)

Written work and the oral examination will both be failed if the oral examination clearly demonstrates that the work was plagiarised. The student is unfamiliar with the arguments of the assignment or the sources used for those arguments.

Plagiarism

Plagiarism is the use of someone else’s work without correct referencing. The consequence of plagiarism is the presentation of someone else’s work as your own work. Plagiarism violates Woolf policy and will result in disciplinary action, but the context and seriousness of plagiarism varies widely. Intentional or reckless plagiarism will result in a penalty grade of zero, and may also entail disciplinary penalties.

Plagiarism can be avoided by citing the works that inform or that are quoted in a written submission. Many students find that it is essential to keep their notes organised in relation to the sources which they summarise or quote. Course instructors will help you to cultivate professional scholarly habits in your academic writing.

Depending on the course, short assignment essays may not require students to submit a bibliography or to use extensive footnotes, and students are encouraged to write their assignments entirely in their own words. However, all essays must acknowledge the sources on which they rely and must provide quotation marks and citation information for verbatim quotes.

There are several forms of plagiarism. They all result in the presentation of someone's prior work as your new creation. Examples include:

- Cutting and pasting (verbatim copying)
- Paraphrasing or rewording
- Unauthorised Collaboration
- Collaboration with other students can result in pervasive similarities – it is important to determine in advance whether group collaboration is allowed, and to acknowledge the contributions or influence of the group members.
- False Authorship (Essay Mills, Friends, and Language Help)
- Paying an essay writing service, or allowing a generous friend to compose your essay, is cheating. Assistance that contributes substantially to the ideas or content of your work must be acknowledged.

Complaints and appeals

Students and faculty should always seek an amicable resolution to matters arising by addressing the issue with the person immediately related to the issue. Students should handle minor misunderstandings or disagreements within a regular teaching session or by direct message, or with their College. If a simple resolution is not possible, or the matter remains unresolved for one party, the steps outlined in this section apply to all groups, colleges, and units of Woolf.

The Red Flag system

An issue with a red flag should be submitted in the case that a member of Woolf seeks to make an allegation of serious misconduct about another member, including matters of cheating, plagiarism, and unfair discrimination or intolerance.

Any member of Woolf, seeking to raise a matter of serious concern, should submit a red flag by emailing redflag@woolf.education. Provide a short, clear description of the issue.

If a student submits an issue with a red flag, or if a faculty member submits an issue about a student, it will trigger a meeting with the student's College Advisor. If the issue is not resolved, the matter will be escalated to the College Dean, or to a committee designated by the College Dean, which will have the power to clear the flag.

If an issue is submitted with a red flag by a faculty member about another faculty member, then the issue is reported directly to the College Dean.

For both students and faculty members, after the Dean's decision, the one who submits the complaint is provided the opportunity to accept or appeal the decision; if the one submitting the issue appeals the decision, it will be assigned to the Quality Assurance, Enhancement, and Technology Alignment Committee, which is a subcommittee of the Faculty Council.

Mitigating circumstances

When serious circumstances ('Mitigating Circumstances'), beyond the control of a student or faculty member, adversely affect academic performance or teaching support, a Mitigating Circumstances report must be submitted using Woolf's red flagging system. Mitigating Circumstances may include but are not limited to serious medical problems, domestic and personal circumstances, major accidents or

interruptions of public services, disturbances during examination, or serious administrative or procedural errors with a material effect on outcomes.

Mitigating circumstances do not normally include a member’s personal technology problems, including software, hardware, or personal internet connection failures; employment obligations or changes in employment obligations; permanent or sustained medical conditions (unless there is a sudden change of condition); or circumstances where no official evidence has been submitted.

Mitigating circumstances are normally only considered when a red flag has been submitted for the issue before the deadline of an affected written project or assignment, or within one week of a cumulative examination. Proof of mitigating circumstances may result in an extended deadline or examination period, or the possibility to retake an examination; it will not result in any regrading of existing submissions or exams.

Grade appeals

Students who dissent from the grades they have received should follow the normal procedure for submitting a red flag.