Module Code	CSU22E03
Module Name	Computer Engineering
ECTS Weighting[1]	5 ECTS
Semester taught	Semester 2
Module Coordinator/s	Prof Merim Dzaferagic
Module Learning Outcomes	On successful completion of this module a student will be able to: LO1. Apply object-oriented programming (OOP) principles, such as encapsulation, inheritance, and polymorphism, to develop efficient and maintainable C++ programs. LO2. Implement and debug C++ programs using object-oriented design, ensuring correctness and efficiency. LO3. Select and apply appropriate data structures and algorithms to solve computational problems effectively. LO4. Integrate software testing techniques, including unit testing and debugging strategies, into the software development process. LO5. Explain how C++ programs are structured and executed at runtime, including memory management and object lifecycle.
Module Content	This module builds on the foundational concepts introduced in an introductory C/C++ programming course, such as the Year 1 Computer Engineering I module. It equips students with the skills to understand and apply object-oriented programming (OOP) principles in C++ to develop practical solutions. Students will design, implement, and debug object-oriented programs using an integrated development environment (IDE). The module also introduces key data structures and algorithms, focusing on their appropriate selection and application in software development.
Teaching and Learning Methods	The module employs a blended teaching approach, combining traditional lectures with interactive coding sessions where the lecturer collaboratively solves problems with students. Weekly hands-on practical sessions provide structured supervision and evaluation, reinforcing key programming concepts. The practical work focuses on the design, implementation, and debugging of fully functional programs, ensuring students develop both theoretical understanding and practical coding skills.

Assessment Details[2]

Assessment Component	Brief Description	Learning Outcomes Addressed	% of total	Week set	Week due
Course Work	Practicals in programming; online quizzes and MCQs	L01-L05	50	Weekly from Week 2	Weekly from Week 2
Examination	Online Realtime 2h	L01-L05	50	n/a	n/a

[1] TEP Glossary
[2] TEP Guidelines on Workload and Assessment

Reassessment Details	Online realtime2h examination (100%)

Contact Hours and Indicative Student Workload

Contact Hours (scheduled hours per student over full module), broken down by:	**55 hours**
Lecture	33 hours
Laboratory	22 hours
Independent study (outside scheduled contact hours), broken down by:	**60 hours**
Preparation for course and review of materials	24 hours
Completion of assessments (including examination, if applicable)	36 hours
Total Hours	**115 hours**

Recommended Reading List

Core Material

- References and online material will be given as the module proceeds.

References for Further Study

- D. S. Malik, C++ Programming: Program Design Including Data Structures, 8th Edition, Cengage Learning, 2018.
- Siddhartha Rao, Sams Teach Yourself C++ in One Hour a Day", 8th Edition, Sams Publishing, 2017.
- Tony Gaddis, Starting Out with C++ from Control Structures to Objects, 9th Edition, Pearsons, 2017.

Advanced Topics

- Bjarne Stroustrup, A Tour of C++, Addison-Wesley, 2018.
- Scott Meyers, Effective C++: 55 Specific Ways to Improve Your Programs and Designs, 3rd Edition, Addison Wesley, 2005.

Module Pre-requisites

Prerequisite modules: CSU11E03.

Other/alternative non-module prerequisites: working knowledge and ability to program in C.

Module Co-requisites

Module Website Material available on black board

Last Update 12/03/2025 by Merim Dzaferagic