Course Title:	**Program Design and Construction**
Course Code:	**COMP603**
Descriptor Start Date:	**01/01/2022**
Descriptor End Date:	**30/01/2023**
POINTS:	**15.00**
LEVEL:	**6**
PREREQUISITE/S:	**COMP503 or COMP610 or ENSE502**
COREQUISITE/S:	**None**
RESTRICTION/S:	**None**

LEARNING HOURS

Hours may include lectures, tutorials, online forums, laboratories. Refer to your timetable and course information in Canvas for detailed information.

Total learning hours: 150

PRESCRIPTOR

An introduction to the design and construction of object- oriented software. It will extend individual design and programming skills developed in earlier programming papers, with an emphasis on the quality, modularity and reusability of the software developed. The paper will introduce current techniques used in software development that allow the goals of software development projects to be realised.

LEARNING OUTCOMES

1. Describe the fundamental issues, concepts and practices associated with software design and construction (a, c)
2. Demonstrate the ability to learn and apply new technical knowledge & skills (d, l)
3. Apply appropriate design techniques to the development of object-oriented software (a, d, e)
4. Assess the quality of software design and implementation (c, e)
5. Explain software reuse as a concept (a)
6. Select and apply appropriate approaches to software reuse (e, c, d)
7. Explain the principles of effective user interface design and apply these to user interface development (a, i, j)
8. Describe the principles and objectives of software testing (c, g)
9. Apply appropriate testing techniques to ensure software quality (a, c, e)

CONTENT

The paper focuses on the fundamental aspects of software design and construction. Whilst the concepts of designing, in-depth understanding of OO concept and writing sound programs may not change. There is a rapid change in the techniques used in the software development process and the technical skills expected. Content will be tailored to meet these changing techniques & skills. In each semester, various technical skills, techniques and the corresponding theoretical aspects will be covered to ensure that students have current knowledge of:
- Object-Oriented Programming principles
- Collections and File input/output
- Object-Oriented Design good practice
- Multi-threading principles
- Graphical User Interface design and Error handling
- Java Database Connectivity
- Design Patterns
- Software Reuse
- Software Quality
- Version Control
- Unit Testing and a testing framework
- Program design and Code Smells

LEARNING & TEACHING STRATEGIES

Participatory teaching methods will be emphasised. Many concepts will be developed through problem-based learning, demonstrations, moderately sized individual programming projects, discussion and analysis. There will also be teacher-directed lectures.

AUT’s Learning Management System will be used to support the students learning in the paper.

ASSESSMENT PLAN

Assessment Event	Weighting %	Learning Outcomes
Software Development Project Assignment 1	40.00	1,2,3,4,5,6
Software Development Project Assignment 2	60.00	1,2,3,4,5,6,7,8,9

Grade Map	**MAP1**
	A+ A A- Pass with Distinction B+ B B- Pass with Merit C+ C C- Pass D Fail

Overall requirement/s to pass the course:

- A minimum mark of 40% in Software Development Project 1 AND
- A minimum mark of 40% in Software Development Project 2 AND
- A minimum of C- (50%) overall grade

LEARNING RESOURCES

No prescribed text.

For further information, contact: Te Ara Auaha - Faculty of Design & Creative Technologies

Principal Programme:	**AK3697, Bachelor of Computer and Information Sciences**
Related Programme/s:	**AK3698** **AK1041** **AK3001** **AK3003** **AK3756** **AK3706**