Course Title:	**Foundation Programming**
Course Code:	**COMP400**
Descriptor Start Date:	**01/01/2023**
Descriptor End Date:	**22/01/2024**
POINTS:	**15.00**
LEVEL:	**4**
PREREQUISITE/S:	**None**
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 structured programming techniques. The student will acquire basic competence in the chosen programming language and will apply this language to simple tasks using sound programming techniques. Includes requirements for problem solving, programme development, testing and documentation.

LEARNING OUTCOMES

1. Recognise and be able to correct the most common syntax errors that occur when writing code in a specified language.
2. Accept user input of a variety of data types, store the data in memory and output it in an appropriate format.
3. Demonstrate the appropriate use of sequence, selection and iteration to design wellstructured programs that adhere to coding standards to carry out simple tasks.
4. Assemble deliberately rearranged code fragments into a working program.
5. Interpret a simple problem specification.
6. Design, write and compile simple programs in the selected language, resolve syntax and logic errors, test and modify as required to meet the original design objectives.
7. Using existing small-scale programs, be able to explain in plain English what their purpose is, and to write test data to show it is working correctly.

CONTENT

Syntax
- Introduce deliberate errors into a Hello World program to become aware of the common syntax errors reported by the compiler.

Code Writing
- Input and store string, integer and decimal data, doing type conversions where necessary.
- Output data in a variety of formats.
- Solve problems that require selection statements to be written.
- Solve problems that require loops to be written.
- Carry out and record tests on solutions, checking expected results against actual results.
- Correct errors detected in code.
- Document code with suitable headers and comments.

Code Understanding
- Examine provided code and predict its outputs.
- Examine provided code and be able to explain its purpose.
- Examine provided code and be able to write test data to show it is working correctly.

Extension
- Investigate more advanced concepts using a variety of software packages.

LEARNING & TEACHING STRATEGIES

- Lab sessions, with feedback through DOMjudge
- Live tutorial sessions on MS Teams where necessary.
- Lectures live and through Panopto videos on CANVAS
- Practical Assessment sessions.
- Discussion forums available through CANVAS

ASSESSMENT PLAN

Assessment Event	Weighting %	Learning Outcomes
Practical Portfolio	50.00	1,2,3,5,6
Assignment	20.00	1,2,3,5,6
Theory Exercise	30.00	1,4,7

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:

- complete the Academic Integrity Module by the end of Week 1 – see below
- attempt and pass the Practical Portfolio;
- attempt and gain a mark of at least 35% in the Theory Exercise;
- achieve a minimum overall grade of C- (50%)

Disclaimer: Course descriptors may be amended between teaching periods/semesters

LEARNING RESOURCES

There is no prescribed text.

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

Principal Programme:	**AK1311, Certificate in Science and Technology**
Related Programme/s:	

Disclaimer: Course descriptors may be amended between teaching periods/semesters