The Hong Kong University of Science and Technology

UG Course Syllabus (Spring 25-26)

Course Title: C Programming Bridging Course

Course Code: COMP 1029C

No. of Credits: 1 Credit

Exclusion(s): COMP 2011, COMP 2012H

Prerequisite(s): COMP 1021 OR COMP 1022P OR COMP 1023 OR ISOM 3230 OR ISOM 3320 OR ISOM 3400

Course Instructor

Name: Gibson Lam

Email: gibson@cse.ust.hk

Course Description

This course introduces the C programming language. It is intended for students who already have some experience in computer programming but wish to learn how to apply those programming skills to the C language. The course covers basic programming topics, such as variables, control, loops, and functions, to more advanced topics. Students explore these by self-learning of course materials together with guided programming exercises. Students without the prerequisites but possess relevant programming knowledge may seek instructor's approval for enrolling in the course. Graded P or F.

List of Topics

Introduction to C
- C Basics
- Variables
- Text Input and Output

Essential Programming in C
- Making Decisions
- For Loops
- While Loops

Functions and Arrays
- Functions
- Variable Scope
- Arrays

Strings and Structures
- Strings
- Structures
- Random Numbers

Advanced Topics
- File Input and Output
- Recursion

Intended Learning Outcomes (ILOs)

By the end of this course, students should be able to:

1. Define basic programming elements such as variables, control, loops and functions in C
2. Describe data structures and data abstraction in C
3. Implement advanced programming concepts in C
4. Design, write and debug computer programs in C

Assessment and Grading

This is a P/F course. P grade will be assigned to students who have achieved the online exercises and final examination passing requirements.

Assessment Task	Passing Requirements
Online exercises	Achieved 60% or higher
Final examination	Achieved 40% or higher

Assessments:

Assessment Task	Contribution to Overall Course Grade (%)	Due date
Online exercises	50%	27/2/2026, 20/3/2026, 17/4/2026 and 8/5/2026
Final examination	50%	TBA

* Assessment marks for individual assessed tasks will be released within two weeks of the due date.

Mapping of Course ILOs to Assessment Tasks

Assessed Task	Mapped ILOs	Explanation
Exercise for Lesson 2	ILO1, ILO4	This task assesses students' ability to define basic programming elements such as variables, control, loops and functions in C (ILO 1), design, write and debug computer programs in C (ILO 4).
Exercise for Lesson 3	ILO1, ILO2, ILO4	This task assesses students' ability to define basic programming elements such as variables, control, loops and functions in C (ILO 1), describe data structures and data abstraction in C (ILO 2), design, write and debug computer programs in C (ILO 4).
Exercise for Lesson 4	ILO1, ILO2, ILO3, ILO4	This task assesses students' ability to define basic programming elements such as variables, control, loops and functions in C (ILO 1), describe data structures and data abstraction in C (ILO 2), implement advanced programming concepts in C (ILO 3), design, write and debug computer programs in C (ILO 4).
Exercise for Lesson 5	ILO1, ILO2, ILO3, ILO4	This task assesses students' ability to define basic programming elements such as variables, control, loops and functions in C (ILO 1), describe data structures and data abstraction in C (ILO 2), implement advanced programming concepts in C (ILO 3), design, write and debug computer programs in C (ILO 4).

Final examination	ILO1, ILO2, ILO3, ILO4	This task assesses students' ability to define basic programming elements such as variables, control, loops and functions in C (ILO 1), describe data structures and data abstraction in C (ILO 2), implement advanced programming concepts in C (ILO 3), design, write and debug computer programs in C (ILO 4).

Grading Rubrics

The following rubrics outline the criteria used to evaluate each ILO, applicable to all assessment tasks involving these ILOs.

Criteria	**Pass**				**Fail**
	Excellent	**Good**	**Satisfactory**	**Needs Improvement**	**Unsatisfactory**
Understanding of Basic Elements	Clearly defines and effectively uses variables, control structures, loops, and functions in C.	Defines and uses basic elements with minor errors.	Basic definitions are present, but application is inconsistent.	Limited definitions and frequent errors in usage.	Fails to define or use basic programming elements.
Data Structures and Abstraction	Provides clear descriptions and effectively uses data structures and abstraction techniques in C.	Describes data structures with minor inaccuracies; some application present.	Basic descriptions of data structures; application is minimal.	Limited understanding of data structures; few applications.	Fails to describe or apply data structures.
Implementation of Advanced Concepts	Effectively implements advanced programming concepts (e.g. recursion) with high proficiency.	Implements advanced concepts with minor errors or inefficiencies.	Basic implementation present but lacks depth or consistency.	Struggles to implement advanced concepts; frequent errors.	Unable to implement advanced programming concepts.
Design, Writing, and Debugging	Designs, writes, and debugs programs with exceptional clarity and efficiency.	Generally designs and writes clear programs; debugging is mostly effective.	Programs are functional but lack clarity; debugging is minimal.	Poor program design and writing; debugging efforts are ineffective.	Fails to design, write, or debug programs.

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
P	Pass	Possesses adequate knowledge of core subject matter, competence in dealing with familiar problems, and some capacity for analysis and critical thinking. Shows persistence and effort to achieve broadly defined learning goals.
F	Fail	Demonstrates insufficient understanding of the subject matter and lacks the necessary problem-solving skills. Shows limited ability to think critically or analytically and exhibits minimal effort towards achieving learning goals. Does not meet the threshold requirements for professional practice or development in the discipline.

Course AI Policy

The use of generative artificial intelligence is not allowed in all assessment tasks.

Communication and Feedback

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission. Feedback on assignments will include detailed scores in various tasks and their grading criteria. Students who have further questions about the feedback including marks should consult the instructor within five working days after the feedback is received.

Resubmission Policy

Resubmission is not permitted for the course.

Required Texts and Materials

N/A

Academic Integrity

Students are expected to adhere to the university's academic integrity policy. Students are expected to uphold HKUST's Academic Honor Code and to maintain the highest standards of academic integrity. The University has zero tolerance of academic misconduct. Please refer to Academic Integrity | HKUST – Academic Registry for the University's definition of plagiarism and ways to avoid cheating and plagiarism.

Additional Resources

Online course content to be published in HKUST canvas when the semester begins.