The Hong Kong University of Science and Technology

UG Course Syllabus (Spring 25-26)

[Course Title] Extended Python Programming Bridging Course

[Course Code] COMP 1028

[No. of Credits] 1 Credit

[Exclusion(s)] COMP 1023, COMP 2011, COMP 2012H

[Prerequisite(s)] COMP 1021

Name: Gibson Lam

Email: gibson@cse.ust.hk

Course Description

This course is designed for students who have already completed COMP 1021 Introduction to Computer Science and want to learn additional material for AI programming in Python, as well as to meet the prerequisites for courses requiring COMP 1023 Introduction to Python Programming. It includes the topics found in COMP 1023 that are not covered in COMP 1021.

The course goes into greater detail on the concepts from COMP 1021 and covers more advanced topics, such as NumPy, Pandas, and Matplotlib. Students will engage in these topics through self-learning of course materials and guided programming exercises. Students without the prerequisites but with relevant programming knowledge may seek the instructor's approval to enroll in the course. The course is graded as Pass or Fail.

List of Topics

- Introduction to Computers and Programming
- Collections
- Functions
- Modularization using modules, packages, and libraries
- Object-Oriented Programming
- NumPy
- Pandas
- Matplotlib

Intended Learning Outcomes (ILOs)

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

1. Describe the in-depth programming mechanisms in Python
2. Apply the core principles of object-oriented programming to create robust, maintainable Python applications
3. Analyze and visualize data using appropriate libraries and tools
4. Write Python code that adheres to industry-standard best practices for coding style and documentation

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
Programming assignment 1	ILO2, ILO3, ILO4	This task assesses students' ability to apply the core principles of object-oriented programming to create robust, maintainable Python applications (ILO 2), analyze and visualize data using appropriate libraries and tools (ILO 3), write Python code that adheres to industry-standard best practices for coding style and documentation (ILO 4).
Programming assignment 2	ILO2, ILO3, ILO4	This task assesses students' ability to apply the core principles of object-oriented programming to create robust, maintainable Python applications (ILO 2), analyze and visualize data using appropriate libraries and tools (ILO 3), write Python code that adheres to industry-standard best practices for coding style and documentation (ILO 4).
Programming assignment 3	ILO2, ILO3, ILO4	This task assesses students' ability to apply the core principles of object-oriented programming to create robust, maintainable Python applications (ILO 2), analyze and visualize data using appropriate libraries and tools (ILO 3), write Python code that adheres to industry-standard best practices for coding style and documentation (ILO 4).
Programming assignment 4	ILO2, ILO3, ILO4	This task assesses students' ability to apply the core principles of object-oriented programming to create robust, maintainable Python applications (ILO 2), analyze and visualize data using appropriate libraries and tools (ILO 3), write Python code that adheres to industry-standard best practices for coding style and documentation (ILO 4).
Final examination	ILO1, ILO2, ILO3	This task assesses students' ability to describe the in-depth programming mechanisms in Python (ILO 1), apply the core principles of object-oriented programming to create robust, maintainable Python applications (ILO 2), analyze and visualize data using appropriate libraries and tools (ILO 3), write Python code that adheres to industry-standard best practices for coding style and documentation (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 Programming Concepts in Python	Demonstrates comprehensive understanding and applies concepts effectively.	Shows good understanding with minor inaccuracies.	Basic understanding; application is inconsistent or lacks depth.	Limited understanding; frequent errors in explanation or usage.	Fails to describe or apply programming concepts.
Application of Object-Oriented Programming Principles	Applies principles to build robust, maintainable applications.	Applies principles with minor design flaws or inefficiencies.	Basic use of principles; design is functional but lacks cohesion.	Struggles to apply principles; code is poorly structured.	Fails to apply object-oriented principles.
Data Analysis and Visualization	Effectively uses tools to analyze and visualize data; insights are clear.	Uses tools with minor errors; analysis and visuals are mostly accurate.	Basic use of tools; analysis or visuals lack clarity or depth.	Limited use of tools; analysis is superficial or visuals are unclear.	Fails to use tools or produce meaningful analysis/visualizations.
Coding Style and Documentation	Code is readable and maintainable; documentation is clear and consistent.	Code generally follows good practices with minor inconsistencies.	Code is functional but lacks consistent style or adequate documentation.	Code shows poor adherence to style; documentation is minimal or unclear.	Code is unreadable, undocumented, and does not follow best practices.

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.