The Hong Kong University of Science and Technology (Guangzhou)

UG Course Syllabus

Course Title: C++ Programming

Course Code: UFUG 2601

Number of credits: 4

Pre-requisites: none

L01

Name: Qiong Luo

Email: luo@hkust-gz.edu.cn

Office Hours: Monday, 2:00PM – 2:50 PM, Friday, 9:30AM-10:20AM, E1-401

L02

Name: Wenye Li and Yuzhe Ma

Email: wenyeli@hkust-gz.edu.cn; yuzhema@hkust-gz.edu.cn

Office Hours:

Yuzhe Ma: Thursday, 9:30AM – 11:30AM, W4-511;

Wenye Li: Monday, 2:00PM – 2:50PM, Friday, 9:30AM – 10:20AM, W4-517.

L03

Name: Jiayi Huang

Email: hjy@hkust-gz.edu.cn

Office Hours: Tuesday and Thursday, 11:00AM – 11:50AM, W1-606

Course Description

This course covers programming and data structures in C++. In addition to basic programming constructs such as variables, arrays, pointers, and functions, students will learn about the standard data structures in C++, such as vectors, sets, maps, and queues. This course will also introduce the basics of object-oriented programming and some useful algorithms in the standard C++ libraries. Weekly laboratory sessions will provide students with hands-on experience in programming.

Intended Learning Outcomes (ILOs)

Upon completion of this course, students are expected to be able to do the following:

1. Explain basic concepts of C++ programming
2. Apply object-oriented approaches to problem solving
3. Use standard data structures and algorithms in C++ programming
4. Write and debug small C++ programs

Weekly schedule & Weekly ILOs

Week	Topics	Weekly ILOs
1	Introduction	Articulate an overview of C++ and the course content
2	Basic syntax, data types, operators	Describe basic language features and write first programs
3	Control flow and functions	Describe core language features and create simple functions
4	Array and vector	Use containers classes, especially array and vector
5	String and Stream	Use C++ strings and iostreams
6	Pointers	Describe memory model, dynamic memory allocation, and use pointers
7	Struct and Classes	Explain basic object-oriented programming concepts and C++ class features, and write classes
8	Inheritance	Re-use C++ classes through inheritance and polymorphism
9	Templates and STL Basics	Use generic programming with C++ templates and the C++ Standard Template Library
10	Stack and Queue	Use the stack and queue container classes in C++
11	Set and Map	Use the set and map container classes in C++
12	Advanced Topics	Get familiar with representative algorithms in C++
13	Course Review	Summarize course content

Assessment and Grading

This course will conduct criterion-referenced assessment. Detailed rubrics for each assignment are provided together with the criteria used for evaluation.

Assessments:

Assessment Task	Contribution to Overall Course grade (%)	Due date
Mid-Term Examination	20%	Around Week 6
Lab Exercises	10%	Distributed across the semester
Programming Assignments	10%+10%+10%	Around Week 4, 8, 12
Final Examination	40%	In university examination period

* 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
Mid-Term Exam	ILO1, ILO2, ILO3	The mid-term exam will be a written exam to assess the students' understanding of C++ core features.
Lab Exercises	ILO1, ILO2, ILO3, ILO4	The lab exercises will reinforce the lecture content and assess students' understanding and programming.
Programming Assignments	ILO2, ILO3, ILO4	The programming assignments will assess the students' object-oriented programming with C++.
Final examination	ILO1, ILO2, ILO3, ILO4	The final exam will be a written exam to assess the students' understanding as well as the practice of C++ programming.

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrates a comprehensive grasp of C++, expertise in problem-solving, and significant creativity in thinking. Exhibits a high capacity for scholarship beyond core requirements to achieve learning goals.
B	Good Performance	Shows good knowledge and understanding of the C++, competence in problem-solving, and the ability to analyze and evaluate issues. Displays high motivation to learn and the ability to work effectively.
C	Satisfactory Performance	Possesses adequate knowledge of core C++, competence in dealing with familiar problems, and some capacity for analysis and critical thinking. Shows persistence and effort to achieve broadly defined learning goals.
D	Marginal Pass	Has threshold knowledge of core C++, potential to achieve key professional skills, and the ability to make basic judgments. Benefits from the course and has the potential to develop in the discipline.
F	Fail	Demonstrates insufficient understanding of C++ 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

Students can use generative AI tools for their research in solving specific programming tasks, but they must be able to explain their complete code for lab exercises and programming assignments. They are encouraged to write their own code to practice C++ programming skills and deepen their understanding.

Communication and Feedback

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission. Feedback on assignments will include specific details, e.g., strengths, and areas for improvement, through grading criteria and test cases. Students who have further questions about the feedback including marks should consult the instructor and the grading TAs within three working days after the feedback is received.

Resubmission Policy

Not applicable

Required Texts and Materials

Recommended Books:

- Stanley Lippman et al., C++ Primer, 2012
- Scott Meyers, Effective Modern C++, 2014
- Paul Deitel, Deitel & Associates, C++ How to Program, 2017

Academic Integrity

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

Additional Resources

Web resources:

- https://cplusplus.com/doc/tutorial/

Online courses:

- Stanford: CS106B Programming Abstractions
- Stanford: CS106L Standard C++ Programming
- Many online courses from Udemy / Coursera / edx / 学堂在线 …