The Hong Kong University of Science and Technology

UG Course Syllabus

Course Title: Honors Object-Oriented Programming and Data Structures

Course Code: COMP2012H

No. of Credits: 5-credit

Any pre-/co-requisites: Grade A or above in COMP 1021 / COMP 1022P / COMP 1022Q (prior to 2020-21) / ISOM 3230

Name: Prof. QUAN Long

Email: quan@cse.ust.hk

Course Description

This course is an accelerated and intensive course on concepts and techniques behind object-oriented programming (OOP) and data structures using an OOP language. It covers the major materials of COMP2011 and COMP2012, and its curriculum is designed for students with excellent programming background or substantial programming experience. Topics include: functions; pointers; abstract data types and their class implementation; static and dynamic construction and destruction of objects; data member and member functions; public interface and encapsulation; class hierarchies; polymorphism; inheritance and dynamic binding; standard template library; generic programming using templates; object-oriented view of data structures: linked lists, queues, stacks, trees, and their algorithms such as searching, sorting and hashing. Prerequisite(s): Grade A or above in
COMP 1021 / COMP 1022P / COMP 1022Q (prior to 2020-21) / ISOM 3230
Exclusion(s): COMP 2011, COMP 2012

List of Topics

1. Introduction to computer programming
2. Fundamentals of C++: data types, variables, operators
3. Flow controls
4. Arrays
5. Functions, scope, and recursion
6. File I/O
7. Pointers
8. lvalue references
9. Structures
10. Class, object construction, initialization, and destruction

11. Separate compilation and makefile
12. Inheritance and polymorphism
13. Generic programming
14. Standard Template Library (STL)
15. Namespace
16. Static data members and member functions
17. rvalue references and move semantics
18. Linked lists
19. Stack and queue
20. Binary search tree
21. AVL tree
22. Hashing

Textbooks and Reference books

Paul Deitel, Deitel & Associates (2017). C++ How to Program.

M.A. Weiss (2014). Data Structures and Algorithm Analysis in C++.

Data Structures and Algorithm Analysis Ed. 3.2 (C++ Version).

Cay S. Horstmann, Wiley (2012). C++ For Everyone.

B. Eckel (2000). Thinking in C++.

L. Nyhoff (2005). ADTs, Data Structures and Problem Solving with C++.

Stanley Lippman (2013). C++ Primer.

Scott Meyers (2005). Effective C++: 55 Specific Ways to Improve Your Programs and Designs.

Grading Scheme

11 Programming assignments	66% (6% each)
Final examination	34%
Total	100%

Course Intended Learning Outcomes

1. Use common software tools to develop and debug a program written in an OOP language.

2. Demonstrate that recursive and non-recursive functions are abstractions of subproblems in a task.
3. Describe the concept and the use of pointers in indirect addressing and dynamic memory allocation.
4. Write object-oriented programs in C++ with object creation, destruction, member variables and functions, inheritance, polymorphisms, and templates.
5. Analyze a program and provide simple solutions with OOP.
6. Write basic algorithms associated with data structures such as stacks, queues, lists, trees, and hashes.
7. Define binary tree and search tree and describe how they are used to solve problems.
8. Develop a large program using separate compilation, good OOP design, and code reuse.