



Xi'an Jiaotong-Liverpool University

西交利物浦大學

***Department of Computer Science and
Software Engineering***

MODULE HANDBOOK

**CPT204
Advanced Object Oriented
Programming**

Erick Purwanto

Semester 2

2020/2021

SECTION A: Basic Information

□ Brief Introduction to the Module

This module will introduce data structures and advanced programming language features within the context of a high-level programming language (Java). It will demonstrate principles, provide indicative examples, develop problem-solving abilities and provide students with experience and confidence in the use of advanced features to implement algorithms in a contemporary software setting. The module is one of the most fundamental modules for all programmes offered at the department. The structure of the module includes lectures and labs.

□ Key Module Information

Module name: Advanced Object Oriented Programming

Module code: CPT204

Credit value: 5

Semester in which the module is taught: 2

Pre-requisites needed for the module: Introduction to Programming in Java, Algorithms and Data Structures

Programmes on which the module is shared: ICS, CST, DMT

□ Delivery Schedule

Lecture room: On-site in SC169 or Online via Zhumu

Lecture times: Tuesday 13:00-15:00, 16:00-18:00

Lab room: Onsite in SD554, SD546 or Online via Zhumu

Lab times: Friday 15:00-16:00, 16:00-17:00, 17:00-18:00

❑ **Module Leader and Contact Details**

Name: Erick Purwanto

Email address: erick.purwanto@xjtlu.edu.cn

Office telephone number: 0512 8188 9179

Room number and office hours: Online via Discussion Forum in Learning Mall or Email for personal matter

Preferred means of contact: Discussion Forum in Learning Mall

❑ **Additional Teaching Staff and Contact Details**

-

SECTION B: What you can expect from the module

❑ **Educational Aims of the Module**

To introduce data structures and advanced programming language features within the context of a high-level programming language (Java). To demonstrate principles, provide indicative examples, develop problem-solving abilities and provide students with experience and confidence in the use of advanced features to implement algorithms in a contemporary software setting.

❑ **Learning Outcomes**

At the end of this module the students should be:

1. Familiar with data structures and advanced programming concepts within Java;
2. Able to carry out the construction of software artefacts utilising these concepts;
3. Capable of carrying out the development of complex elements, such as user interfaces, multiprocessing, and fault tolerant components.

❑ Assessment Details

Initial Assessment

Sequence	Method	Assessment Type (EXAM or CW)	Learning Outcomes Assessed	Duration	Week	% of Final Mark	Resit (Y/N/S)
01	Exam	EXAM	All	2 hours	16	50	S
02	Continuous Assessment	CW	All		14	50	S

Resit Assessment

Sequence	Assessment Type (EXAM or CW)	Learning outcomes assessed (use codes under Learning Outcomes)	Duration	Week	% of Final Mark
R01	Resit Exam	All	2 hours	Resit	100

The details of Exam and Continuous Assessment will be announced later in the semester. The resit exam will assess all of the learning outcomes of the module, and will be weighted as 100% of the final module mark. Other components of the assessment, regardless of whether or not the student passed or failed, will not be included in the calculation of the final module mark, following the resit exam.

❑ Methods of Learning and Teaching

In this hyflex / on-site & online delivery and assessment period, students will be expected to study the lecture and lab materials, and complete all the assessment components either in the lecture/lab room or within the comfort of their homes online. Lectures will introduce students to the Java object-oriented techniques, data structures and their applications, which are the subject of the module.

Furthermore, there will be weekly online programming exercises that will allow students to practice those skills. The coursework will conclude in a lab exam that will be used to test to what extent practical skills have been learnt.

□ Tentative Teaching Plan

Week	Date	Teaching Content	Remark
01	1 Mar 2021	Introduction, Course Admin, Basic Java Review, Checking, Testing	
02	8 Mar 2021	Checking and Testing 2, Immutability, List, Map	
03	15 Mar 2021	Coding Rules, Testing 3, Recursion	
04	22 Mar 2021	Testing 4, Recursive Linked List	
05	29 Mar 2021	Generics, Linked List, Linked-based Deque	
06	5 Apr 2021	Specification, Exception, Deep Copy, Copy Constructor	
07	12 Apr 2021	Midterm Break	
08	19 Apr 2021	Array-based List, Resizing Array, Circular Array, Array-based Deque	
09	26 Apr 2021	Disjoint Sets, Immutability, Defensive Programming, Iterable	
10	3 May 2021	ADT, Interface, Inheritance, Dynamic Method Selection, Set	
11	10 May 2021	Invariant, Abstraction Function, Equals, Comparable	
12	17 May 2021	Comparator, Map, Hash Code, Hash Table, Concurrency	
13	24 May 2021	Networking, Message Passing, Synchronization, Deadlock, Priority Queues	
14	31 May 2021	Conclusion and Review	Coursework in Lab

□ Reading Materials

Required (Essential) Textbook: -

Recommended Texts: Effective Java by Joshua Bloch

Additional Readings: -

SECTION C: Further Information

❑ Attendance

Students who are able to be on campus are reminded of the Academic Policy requiring no less than 80% attendance at classes. Failure to observe this requirement may lead to failure or exclusion from resit examinations or retake examinations in the following year.

❑ Student Feedback

The University is keen to elicit student feedback to make improvements for each module in every session. It is the University policy that the preferred way of achieving this is by means of an Online Module Evaluation Questionnaire Survey. Students will be invited to complete the questionnaire survey for this module at the end of the semester.

You are strongly advised to read the policies mentioned below very carefully, which will help you better perform in your academic studies. All the policies and regulations related to your academic study can be found in 'Student Academic Services' section under the heading "Policies and Regulations" on [E-bridge](#).

❑ Plagiarism, Cheating, and Fabrication of Data.

Offences of this type can result in attendance at a University-level committee and penalties being imposed. You need to be familiar with the rules. Please see the "Academic Integrity Policy" available on e-Bridge in the 'Student Academic Services' section under the heading 'Policies and Regulations'.

❑ Rules of submission for assessed coursework

The University has detailed rules and procedures governing the submission of assessed coursework. You need to be familiar with them. Details can be found in the "Code of Practice for Assessment" available on e-Bridge in the 'Student Academic Services' section under the heading 'Policies and Regulations'.

❑ Late Submission of Assessed Coursework

The University attaches penalties to the late submission of assessed coursework. You need to be familiar with the University's rules. Details can be found in the "Code of Practice for Assessment" available on e-Bridge in the 'Student Academic Services' section under the heading 'Policies and Regulations'.

❑ Mitigating Circumstances

The University is able to take into account mitigating circumstances, such as illness or personal circumstances which may have adversely affected student

performance on a module. It is the student's responsibility to keep their Academic Advisor, Programme Director, or Head of Department informed of illness and other factors affecting their progress during the year and especially during the examination period. Students who believe that their performance on an examination or assessed coursework may have been impaired by illness, or other exceptional circumstances should follow the procedures set out in the "Mitigating Circumstances Policy", which can be found on e-Bridge in the 'Student Academic Services' section under the heading 'Policies and Regulations'.

❑ **Learning Mall**

Copies of lecture notes and other materials are available electronically through Learning Mall, the University's virtual learning environment.

.