Lingnan University

Department of Computing and Decision Sciences

Course Syllabus

Course Title	:	Data Structures and Object-oriented Programming
Course Code	:	CDS2003
Recommended Study Year	:	2
No. of Credits/Term	:	3
Mode of Tuition	:	Sectional Approach
Class Contact Hours	:	3 hours per week
Category in Major Prog.	:	Required
Prerequisite(s)	:	CDS1001 Introduction to Programming for Data Science
Co-requisite	:	Nil
Exclusion	:	Nil
Exemption Requirement	:	Nil

Brief Course Description:

Computer algorithms manipulate data. A good data structure design allows algorithms to perform efficiently. Object-oriented programming is a programming paradigm based on the concepts of “objects”, which are data structures that can exhibit “behaviour”. Students will learn the basics of data structures for various algorithms, implement language models for generative artificial intelligence (AI) and design software using object-oriented programming.

Aims:

This course aims at introducing concepts of abstract data types and object-oriented programming. Abstract data structures and classes, such as vector, list, stack, queue, tree, trie, set, and hashing, will be discussed. Topics in object-oriented programming include abstraction, encapsulation, inheritance and polymorphism. Additionally, the course will cover the basics of language models, including n-gram models for understanding the generative AI. These concepts will be illustrated with sample algorithms, such as binary search, interpolation search, and sorting, in Python or Java.

Learning Outcomes (LOs):

Upon the successful completion of this course, the student will be able to:

1. Identify the basic concepts about complexity analysis in algorithms;
2. Identify and implement various data structures;
3. Describe, implement, and apply different search and sorting algorithms;
4. Recall and apply in-depth knowledge about object-oriented programming;
5. Describe and implement the algorithms of language models for generative AI;
6. Apply critical thinking skills to analyse problems; (PLO3)
7. Formulate problems creatively and solve them using different algorithms; (PLO4)
8. Implement programs based on various data structures and object-oriented programming. (PLO9)

Indicative Contents:

Analysis in Algorithms
Basic concepts of complexity measures
Recursion

Basic Data Structures
Vector, linked list, stack, queue, dequeue, tree, trie, set, and hashing

Search Algorithms
Linear search for unordered/ordered lists, binary search, and interpolation search

Sorting Algorithms
Insertion sorts, exchange sort, selection sort, merge sort, quicksort, bucket sort, radix sort, and topological sort

Language Model for Generative AI
Types of language models (n-gram models, neural network-based models, transformer-based models), applications of language models (text generation, text classification, sentiment analysis), ethical considerations in using language models

Object-oriented Programming
Abstract classes and abstract methods, encapsulation, superclass and subclass , inheritance, and polymorphism

Teaching Method:

There are a number of teaching and learning activities including lectures, laboratories and group projects. The concepts and principles of complexity analysis in algorithms, data structures, search algorithms, sort algorithms, language models and object-oriented programming will be covered in lectures. The implantations of data structures, algorithms, and language models in an object-oriented language will be taught during the laboratories. Students are required to perform a group project to apply the concepts and principles covered in this course to critically analyse the given problem(s) and creatively formulate the solution(s). Students implement the solution(s) in an object-oriented language.

Measurement of Learning Outcomes:

	Class Attendance and Participation	Assignments	Group Project	Final Examination
1. Identify the basic concepts about complexity analysis in algorithms	x			x
2. Identify and implement various data structures	x			x
3. Describe, implement, and apply different search and sorting algorithms		x		x
4. Recall and apply in-depth knowledge about object-oriented programming		x		x
5. Describe and implement the algorithms of language models for generative AI;				x
6. Apply critical thinking skills to analyse problems			x	
7. Formulate problems creatively and solve them using different algorithms			x	
8. Implement programs based on various data structures and object-oriented programming		x	x	

1. There are a number of classroom activities to evaluate the basic understanding on analysis in algorithms and properties of various data structures (LO1-2).
2. Assignments require students to develop object-oriented programs to implement various data structures and algorithms (LO3-4, LO8, and PLO9).
3. A Group Project requires students to apply the right data structures and algorithms to develop program(s) in an object-oriented programming language. They need to critically analyse the problem(s) and creatively formulate the solution(s). The solution(s) will then be implemented in the program(s) (LO6-8, PLO3, PLO4, PLO9).
4. The understanding of data structures, algorithms, language models, and object-oriented programming is evaluated in the examination (LO1-5).

Assessment:

Class Attendance and Participation	5%
Assignments	35%
Group Project	20%
Final Examination	40%
Total	100%

Required/Essential Readings:

1. Jain, Hermant. *Problem Solving in Data Structures and Algorithms Using Python: Programming Interview Guide*. CreateSpace Independent Publishing Platform, 2016.
2. Ramalho, Luciano. *Fluent Python: Clear, Concise, and Effective Programming*. O'Reilly Media, 2015.
3. Jurafsky, Daniel, and James H. Martin. Speech and Language Processing (3rd ed.). Draft of the 3rd edition, 2019.

Recommended/Supplementary Readings:

1. Aho, Alfred V., Ullman, Jeffrey D., and Hopcroft, John E. *Data Structures and Algorithms*. Pearson, 1983.
2. Goodrich, Michael T., Tamassia, Roberto, and Goldwasser, Michael H. *Data Structures and Algorithms in Java*. 6th Edition. Wiley, 2014.
3. Goodrich, Michael T., Tamassia, Roberto, and Goldwasser, Michael H. *Data Structures and Algorithms in Python*. Wiley, 2016.
4. Karumanchi, Narasimhi. Data Structures and Algorithms Made Easy: *Data Structures and Algorithmic Puzzles*. 5th Edition, CareerMonk Publications. 2016.
5. Liang, Y. Daniel. *Introduction to Java Programming, Comprehensive Version*. 10th Edition. Pearson, 2014.

Important Notes:

(1) Students are expected to spend a total of 9 hours (i.e. 3 hours of class contact and 6 hours of personal study) per week to achieve the course learning outcomes.

(2) Students shall be aware of the University regulations about dishonest practice in course work, tests and examinations, and the possible consequences as stipulated in the Regulations Governing University Examinations. In particular, plagiarism, being a kind of dishonest practice, is "the presentation of another person's work without proper acknowledgement of the source, including exact phrases, or summarised ideas, or even footnotes/citations, whether protected by copyright or not, as the student's own work". Students are required to strictly follow university regulations governing academic integrity and honesty.

(3) Students are required to submit writing assignment(s) using Turnitin.

(4) To enhance students' understanding of plagiarism, a mini-course "Online Tutorial on Plagiarism Awareness" is available on https://pla.ln.edu.hk/.

Rubric for Final Examination of CDS2003 - Data Structures and Object-Oriented Programming

Criteria	Very good (4-6)	Satisfactory (2-4)	Unsatisfactory (0-2)
Recongize the basic concepts about complexity analysis in algorithms (LO1)	Given the complexity of different algorithms, the student can correctly identify the efficiencies of most of the algorithms in different situations. Explanations have been given and most of them are correct.	Given the complexity of different algorithms, the student can correctly identify the efficiencies of some algorithms in different situations. Explanations have been given but some of them may be incorrect.	Given the complexity of different algorithms, the student can correctly identify the efficiencies of very few of these algorithms. Explanations have not been provided or the explanations are basically incorrect.
Recognize the properties of various data structures (LO2)	The descriptions are completely correct as well as being clear and concise.	The descriptions are nearly correct, although not so clear or concise.	The descriptions are basically incorrect.
Implement various data structures (LO2)	The implementations are correct, simple, and straight forward.	The implementations are nearly correct. Unnecessary codes are used in the implementations.	The implementations are basically incorrect.
Describe various search and sorting algorithms (LO3)	The student demonstrates a clear understanding of these algorithms. The sThe student can elaborate nearly all of their properties. The elaborations are completely / nearly completely correct and precise.	The student demonstrates a reasonable understanding of these algorithms. The sThe student can elaborate most of their properties. Some elaborations are not completely correct or precise.	The student demonstrates limited understanding of these algorithms. The sThe student can only elaborate few, or any, algorithms.
Implement and apply various search and sorting algorithms (LO3)	The student can correctly select the right algorithm(s) to handle the given problem(s). The implementations are correct, simple, and straight forward.	The student can correctly select the right algorithm(s) to handle the given problem(s). The implementations are nearly correct. Unnecessary codes are used in the implementations.	The student selects the wrong algorithm(s) to handle the given problem(s), or the implementations are basically incorrect.
Recall knowledge about object-oriented programming (LO4)	The student demonstrates a clear understanding of object-oriented programming. The student can elaborate nearly all of its concepts. The elaborations are completely / nearly completely correct and precise.	The student demonstrates a reasonable understanding of object-oriented programming. The student can elaborate most of its concepts. Some elaborations are not completely correct and precise.	The student demonstrates limited understanding of object-oriented programming. The student can only elaborate very few of its concepts.
Describe and implement the algorithms of language models for generative AI (LO5)	The student can implement and describe the language models correctly. The student can elaborate nearly all types of language models. The implementations are correct, simple, and straight forward.	The student can implement and describe the language models reasonably. The student can elaborate some types of language models. The implementations are nearly correct.	The student can implement and describe the language models incorrectly. The student can elaborate limited types of language models. The implementations are incorrect.

Rubric for Individual Assignments of CDS2003 - Data Structures and Object-Oriented Programming

Criteria	Very good (4-6)	Satisfactory (2-4)	Unsatisfactory (0-2)
Implement various data structures (LO7)	The student can correctly select the right data structure(s) to handle the given problem(s). The implementations are correct, simple, and straight forward.	The student can correctly select the right data structure(s) to handle the given problem(s). The implementations are nearly correct. Unnecessary codes are used in the implementations.	The student select the wrong data structure(s) to handle the given problem(s), or the implementations are basically incorrect.
Implement and apply various search and sorting algorithms (LO3)	The student can correctly select the right algorithm(s) to handle the given problem(s). The implementations are correct, simple, and straight forward.	The student can correctly select the right algorithm(s) to handle the given problem(s). The implementations are nearly correct. Unnecessary codes are used in the implementations.	The student selects the wrong algorithm(s) to handle the given problem(s), or the implementations are basically incorrect.
Efficiency of the implemented programs (LO3)	The implemented programs are efficient.	The implemented programs are quite efficient.	The programs cannot be implemented, or the programs are inefficient.
Correctness of the outputs of the programs (LO3)	The implemented programs generate correct outputs.	Some outputs of the implemented programs are correct.	The programs cannot be implemented, or the programs produce incorrect outputs.
The Object-Oriented Programming (OOP) features used in the implemented programs (LO4)	Many OOP features are used appropriately.	Some OOP features are used appropriately	The programs cannot be implemented, or very few OOP features are used.

Rubric for Group Project of CDS2003 Data Structures and Object-Oriented Programming

Criteria	Very good (4-6)	Satisfactory (2-4)	Unsatisfactory (0-2)
Problem definition (demonstrate the ability in applying critical thinking skills to analyse problems and formulating alternative solutions) **(LO5)**	Clearly state the problem, list out related constrains, critically analyse the problem, and are able to formulate alternative solutions.	The problem is stated but related constrains, critical analysis, and alternative solutions are not considered thoroughly.	The problem is marginally defined and with no or little consideration of constrains, critical analysis, and alternative solutions.
Creative solution design	The design of the solution is strongly related to the problem and the approaches and techniques involved are clearly explained. Moreover, the design of the solution is very creative.	The design of the solution is related to the problem but the approach and techniques involved are not clearly explained.The design of the solution is not so creative.	The design of the solution is weakly related to the problem and no / little / unclearexplanation of the approach and technique is involved. The design of the solution is not creative.
Implement various data structures **(LO7)**	Students can correctly select the right data structure(s) for the solution. The implementations are correct, simple, and straight forward.	Students can correctly select the right data structure(s) for the solution. The implementations are nearly correct. Unnecessary codes are used in the implementations.	Students select the wrong data structure(s) for the solution, or the implementations are basically incorrect.
Efficiency of the solution **(LO7)**	The implemented solution is efficient.	The implemented solution is quite efficient.	The solution cannot be implemented, or the solution is inefficient.
Correctness of the solution **(LO7)**	The solution is correct.	The solution is mostly correct.	The solution is far from correct.
Have students used the Object-Oriented Programming (OOP) paradigm **(LO7)**	The solution is implemented using OOP paradigm.	The solution is implemented with some OOP features.	The solution is implemented with few or no OOP features.

Rubric for Class Attendance and Participation of CDS2003 Data Structures and Object-Oriented Programming

Criteria	Very good (4-6)	Satisfactory (2-4)	Unsatisfactory (0-2)
Attendance	Full, punctual attendance in class and mandatory seminars.	Occasional absences or lateness from class or mandatory seminars.	Frequent or recurring absence or lateness from class or mandatory seminars.
Quality of comments	Comments are always insightful & constructive. Comments balanced between general programming knowledge, opinions & specific, thoughtful criticisms or contributions.	Comments are sometimes constructive, with occasional signs of insight. Comments not always relevant to the discussion.	Comments are uninformative. Heavy reliance on personal opinion & personal feeling
Participate in class discussion	The students is always being active during class discussions.	The student is sometimes being active during class discussions.	The student keep silent during class discussions.
Enthusiasm in the subject	The student always respond to teacher's questions, or raise questions or provide further examples related to the topic of interest.	The student sometimes respond to teacher's questions, or raise questions, or provide further examples related to the topic of interest.	The student seldom or never respond to teacher's questions, nor raise questions, nor provide further examples related to the topic of interest.