浙江大學

COMP523

Data Structures & Algorithms

Instructor contact details

Lecturer-in-charge: TBA
Email: TBA or send general enquiries to wlwyxy_29@zju.edu.cn
Office location: to be announced
Consultation Times: to be announced, and by appointment

Teaching Times and Locations

The Time and Location: TBA, Zhejiang University Huajiachi Campus
Lecture sessions include lectures, as well as in-class discussions and test(s).

Academic Level
Undergraduate

Units of Credit

The course is worth 6 units of credit Zhejiang University, with total of 40 teaching hours.

Overview

The goal of this course is to deepen your understanding of data structures and algorithms and how these can be employed effectively in the design of software systems. It is an important course in covering a range of core data structures and algorithms that will be used in context in later courses. You explore these ideas in lectures, tutorials, lab classes, and assignments. Assessment involves labs, tutorials, practical lab exams, a practical final exam, and a theory exam. At the end of the course, we want you to be a solid programmer, with knowledge of a range of useful data structures and programming techniques, capable of building significant software systems in a team environment, and ready to continue with further specialised studies in computing.

Prerequisites

None

Course Aims

The aim of this course is to get you to think like a computer scientist. This certainly sounds like a noble goal... but what does it really mean? How does a scientist, let alone a computer scientist, actually think? What many types of scientists try to do is understand natural systems and processes: a geologist, for example, tries to understand the structure of the earth; a biologist tries to understand living organisms; a chemist tries to understand materials and reactions, and so on.

Computer scientists don't, as the name might suggest, simply try to understand the structure and behaviour of computers but are more concerned with understanding software systems (and the interaction between the software and the hardware on which it runs). Also, unlike other scientists, computer scientists frequently build the objects that they study. During this course, we'll be looking at ways of creating, analysing and understanding software. Ultimately, you should be able to answer the question, is this piece of software any good? and be able to provide sound reasons to justify your answer.

Student Learning Outcomes

On successful completion of this course students will be able to:

1	Skills in performing analysis of given recursive and iterative algorithms.
2	Understanding and performing simple proofs of algorithmic complexity and correctness.
3	An ability to understand and derive recurrences describing algorithms and properties of data structures.
4	An understanding of the implementation and efficiency of a range of data structures including, trees, binary heaps, hash-tables and graphs.
5	An understanding of a variety of well-known algorithms on some of the data structures presented.
6	The ability to implement and use these algorithms in code.
7	A foundational understanding of intractability. An understanding of proof techniques for NP-Completeness.
8	An ability to solve new analytic and algorithmic problems.

Teaching Strategies

Involves lectures, tutorials, labs, assignments and a text book.
Lectures aim to convey basic information about the course content and to model the practices and techniques involved in software development (i.e., we do demos). The most important components of the course, however, are the tutorials, labs and assignments. Tutorials aim to clarify and refine the knowledge that you got from lectures, and from reading the text book and notes. Labs and assignments are where you get to put together and practice all of the ideas from the lectures, and text. The only way to develop the skills to do effective software development is by practicing them.

The University requires us to assess how well you have learned the course content, and the primary approach to achieving this is via a final exam. An exam is the ultimate summative assessment tool; it gives you a chance, at the end of the course, to demonstrate everything that you've learned. Labs and assignments are a learning tool, not an assessment tool, so, in an ideal world, we would have them as pure learning exercises and award no marks for them. However, to give a more concrete incentive to do them (in a timely fashion), there are marks tied to them.

Required Resources

The textbook for this course is Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, MIT Press.

Recommended further reading:

1. Algorithms and Data Structures - The Basic Toolbox by Kurt Mehlhorn and Peter Sanders, Springer, 2008. (the full text is available on the Author's website).
2. Data Structures and Algorithms in Java by Michael T. Goodrich, Irvine Roberto Tamassia, and Michael H. Goldwasser, Wiley, 6th Edition, 2014. (available in the library).

Course Delivery

The course will be taught in English through lectures, seminars, field trips, group activities. In addition to these, there will also be guest speakers and optional field trips available for students who would like to enhance their learning experience. The course sessions will be running during the weekdays, Monday to Friday. The course will be at a total of 40 hours.

Schedule

Week	Description	
Day 1	Introduction to the course Analysis of Algorithms	23 / 12 / 2019
Day 2	Analysis of ADT (multiple) implementations Trees, Binary Search Trees (BST)	24 / 12 / 2019
Day 3	Balanced Trees, Search Tree Algorithms	25 / 12 / 2019
Day 4	Graph ADT, Graph Algorithms (1)	26 / 12 / 2019
Day 5	Graph Algorithms (2)	27 / 12 / 2019
Day 6	Mid-term examination	28 / 12 / 2019
Day 7	Hashing, Heaps	29 / 12 / 2019
Day 8	Generic ADTs in C Sorting (1)	02 / 01 / 2020
Day 9	Sorting (2)	03 / 01 / 2020
Day 10	LABS	06 / 01 / 2020
Day 11	Text processing algorithms	07 / 01 / 2020
Day 12	Course Review	08 / 01 / 2020
Day 13	Review Exercises	09 / 01 / 2020
Day 14	Assessment Due: Final exam	10 / 01 / 2020

Assessments

Details on each of the assessments will be discussed during class lectures.

Assignment	30%	Due date: 09 / 01 / 2020
Mid-term examination	30%	In class: 28 / 12 / 2019
Labs	10%	In class: 06 / 01 / 2020
Final exam	30%	In class: 10 / 01 / 2020

Grade Descriptors

HD	High Distinction	85 - 100
D	Distinction	75 - 84
CR	Credit	65 - 74
P	Pass	50 - 64
F	Fail	0 - 49

High Distinction 85-100

Treatment of material evidences an advanced synthesis of ideas
Demonstration of initiative, complex understanding and analysis
Work is well-written and stylistically sophisticated, including appropriate referencing, clarity, and some creativity where appropriate
All criteria addressed to a high level

Distinction 75-84

Treatment of material evidences an advanced understanding of ideas Demonstration of initiative, complex understanding and analysis Work is well-written and stylistically strong
All criteria addressed strongly

Credit 65-74

Treatment of material displays a good understanding of ideas
Work is well-written and stylistically sound, with a minimum of syntactical errors All criteria addressed clearly

Pass 50-64

Treatment of material indicates a satisfactory understanding of ideas Work is adequately written, with some syntactical errors
Most criteria addressed adequately

Fail 0-49

Treatment of ideas indicates an inadequate understanding of ideas Written style inappropriate to task; major problems with expression Most criteria not clearly or adequately addressed

Academic Integrity

Students are expected to uphold the university's academic honesty principles which are an integral part of the university's core values and principles. If a student fails to observe the acceptable standards of academic honesty, they could attract penalties and even disqualification from the course in more serious circumstances. Students are responsible for knowing and observing accepted principles of research, writing and any other task which they are required to complete.

Academic dishonesty or cheating includes acts of plagiarism, misrepresentation, fabrication, failure to reference materials used properly and forgery. These may include, but are not limited to: claiming the work of others as your own, deliberately applying false and inaccurate information, copying the work of others in part or whole, allowing others in the course to copy your work in part or whole, failing to appropriately acknowledge the work of other scholars/authors through acceptable referencing standards, purchasing papers or writing papers for other students and submitting the same paper twice for the same subject.

This Academic Integrity policy applies to all students of the Zhejiang University in all programmes of study, including non-graduating students. It is to reinforce the University's commitment to maintain integrity and honesty in all academic activities of the University community.

Policy

The foundation of good academic work is honesty. Maintaining academic integrity upholds the standards of the University.

The responsibility for maintaining integrity in all the activities of the academic community lies with the students as well as the faculty and the University. Everyone in this community must work together to ensure that the values of truth, trust and justice are upheld.

Academic dishonesty affects the University's reputation and devalues the degrees offered.

The University will impose serious penalties on students who are found to have violated this Policy. The following penalties may be imposed:

- Expulsion;
- Suspension;
- Zero mark/fail grade;
- Marking down;
- Re-doing/re-submitting of assignments or reports; and o Verbal or written warning.