Handbook Description

At the core of most computer applications is the storage and retrieval of information. The way that the stored data is structured has a strong impact on what can be retrieved, how quickly it can be retrieved, and how much space it occupies. The use of generic structures, or abstract data types (ADTs), to encapsulate the data also allows software engineering principles of independent modification, extension and reuse.

This unit studies the specification, implementations and time and space performance of a range of commonly-used ADTs and corresponding algorithms in an object-oriented setting. The aim is to provide students with the background needed both to implement their own ADTs where necessary, and to select and use appropriate ADTs from object-oriented libraries where suitable.
Unit Outcomes

At the end of the unit you will:

1. be able to programme proficiently in Java.
2. be able to identify and abstract computational problems.
3. be able to identify, analyse and implement and apply a range of common data structures.
4. know important algorithmic techniques and a range of useful algorithms.
5. be able to implement algorithms as a solution to any solvable problem.
6. be able to analyse the complexity and correctness of algorithms and data structures.
7. be able to design correct and efficient algorithms.

The course will proceed by covering a number of algorithms; as they are covered, the general algorithmic technique involved will be highlighted, and the role of appropriate data structures, and efficient implementation considered.
Timetable

• Lectures
  – 3pm-4pm Tuesdays, Austin Lecture Theatre
  – 8am-9am Thursdays, Austin Lecture Theatre

• Workshops
  – 9am-10am Thursdays, Austin Lecture Theatre

• Laboratories
  – 4pm-6pm Tuesdays, CSSE Lab 2.03
  – 8am-10pm Wednesdays, CSSE Lab 2.03
  – 10am-12pm Wednesdays, CSSE Lab 2.03
  – 12pm-2pm Wednesdays, CSSE Lab 2.03

• Consultation
  – 2pm-3pm Friday, CSSE Room 1.07

You may also get help via the help2200 electronic forum — a public discussion board for all queries relating to the unit.
Assessment

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Date</th>
<th>% of Final Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laboratory work</td>
<td>Starting Week 2</td>
<td>20%</td>
</tr>
<tr>
<td>Mid-semester test</td>
<td>Thursday March 23, Week 4</td>
<td>10%</td>
</tr>
<tr>
<td>Project</td>
<td>Weeks 9-13</td>
<td>20%</td>
</tr>
<tr>
<td>Final examination</td>
<td>June examination period</td>
<td>50%</td>
</tr>
</tbody>
</table>

References

Further Reading

There are many different books on the subject of data structures as well as books on the subject of Java, including some which combine the two. A few examples of books worth looking at include:

Topics Of Study

We will study the following topics this semester:

<table>
<thead>
<tr>
<th>1. Intro to Data Structures</th>
<th>2. Intro to Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Stacks and Queues</td>
<td>4. Data Abstraction</td>
</tr>
<tr>
<td>5. Lists</td>
<td>6. Complexity Analysis</td>
</tr>
<tr>
<td>7. Objects and Iterators</td>
<td>8. Trees and Graphs</td>
</tr>
</tbody>
</table>
What You Should Do This Week

1. Get set up to use the School’s Computer Systems:
   https://secure.csse.uwa.edu.au/run/csentry?pw1=yes

2. Begin to familiarise yourself with the Unit’s web site.

3. Work through An Introduction to MacOSX and the first labsheet.

4. Revise Java.

   Laboratory and tutorial classes start next week!