

Module Information

Module Identifier	CS21120
Module Title	Program Design, Data Structures and Algorithms
Academic Year	2014/2015
Co-ordinator	<u>Dr Bernard Paul Tiddeman (mailto:bpt@aber.ac.uk?subject=CS21120)</u>
Semester	Semester 2 (Taught over 2 semesters)
Mutually Exclusive	<u>CC21120 (?m=CC21120)</u>
Pre-Requisite	<u>CS12420 (?m=CS12420)</u>
Other Staff	<u>Dr Harry George Strange (mailto:hgs08@aber.ac.uk?subject=CS21120)</u> <u>Dr Lynda Ann Thomas (mailto:ltt@aber.ac.uk?subject=CS21120)</u> <u>Dr Bernard Paul Tiddeman (mailto:bpt@aber.ac.uk?subject=CS21120)</u> <u>Dr Yonghuai Liu (mailto:yyl@aber.ac.uk?subject=CS21120)</u> <u>Dr Andrew John Starr (mailto:aos@aber.ac.uk?subject=CS21120)</u> <u>Mr Neil Seosamh Macparthalain (mailto:ncm@aber.ac.uk?subject=CS21120)</u> <u>Dr Neal Snooke (mailto:nns@aber.ac.uk?subject=CS21120)</u>

Course Delivery

Delivery Type	Delivery length / details
Lecture	44 lectures

Assessment

Assessment Type	Assessment length / details	Proportion
Semester Assessment	There will also be regular worksheets with penalties for non-submission. Course Work: Two pieces.	50%
Semester Exam	2 Hours Written examination	50%
Supplementary Assessment	2 Hours Resit failed examination and/or resubmission of failed/non-submitted coursework components or ones of equivalent value.	100%
Supplementary Exam	2 Hours supplementary exam	50%

Learning Outcomes

On successful completion of this module, students should be able to:

- demonstrate their understanding of the principles of abstraction and encapsulation as they apply to the design of abstract data types and programs (A1, A2);
- analyse and evaluate the time and space behaviour of algorithms and understand how this is expressed and determined (A1, A2, A3);
- recognise the importance of this analysis in the design of software (A1, A2);
- recognise the importance of the classes P and NP in the analysis of algorithms (A1);
- describe some of the main approaches to algorithm design such as greedy algorithms, divide and conquer and dynamic programming (A1);
- demonstrate judgement in evaluating and choosing appropriate data structures and algorithms for a range of programming problems (A1, A2);
- design and implement significant programs in Java (A2, A3).

Brief description

This module builds on the foundations of the first year modules on program design and provides a thorough grounding in the design of data structures and algorithms and gives further insight into object-oriented design.

Aims

This module provides an introduction to data structures and their use in solving programming problems. The module emphasises the use of abstract data types and the contribution that abstraction and encapsulation can make to the comprehensibility, reusability and robustness of programs. The module also examines the efficiency of well-known algorithms in order to provide a basis for students to make informed choices about data structures and algorithms. Java is used as the language of implementation with the intent of providing a means of allowing the student to naturally express these design objectives in code.

The module is also concerned with the reuse of software design patterns and frameworks, thereby reducing the need to build programs from first principles.

As well as providing a solid grounding in the major data structures and algorithms of Computer Science, the module stresses the development of problem solving skills through a number of programming worksheets.

Content

1. Module Overview - 1 Lecture

An overview of the method of teaching and assessment, and a road-map of the topics to be covered and their relationships. Some basic concepts are introduced.

2. Program design issues - 4 Lectures

Explanation of design issues such as object-orientation and identification of components through case-study examples.

3. Design patterns and frameworks - 5 Lectures

An introduction to object-oriented design patterns and frameworks. Support for reuse. General concepts, representation and examples. How patterns may be implemented in Java.

4. Introduction to Complexity - 2 Lectures

O() notation, growth rates. Measurement of execution time of some example programs and estimation of their time complexity. P and NP.

5. Classes of Algorithm - 2 Lectures

An overview will be given on the different classes of algorithm; for example, divide and conquer and greedy algorithms.

6. Recursion - 2 Lectures

An introduction to recursive thinking. Examples of recursion.

7. Storing and Retrieving Data by Key (1) - 13 Lectures

This problem will be used to motivate the discussion of a wide variety of different implementation techniques. The features of some typical solutions will be related to the dimensions of the problem such as the volume of data to be handled, volatility and the operations required. Internal Storage: linear and binary searching. Linked representations; an introduction to hashing, binary search trees, AVL trees and heaps.

8. Storing and Retrieving Data by Key (2): External storage - 4 Lectures

Performance issues. Hashing and B-tree organisations. The Hashable class in Java.

9. Sorting - 4 Lectures

A comparison of divide and conquer, priority queue and address calculation based sorting algorithms. Performance characteristics of these algorithms will be discussed.

10. Representing Complex Relationships: Graphs - 7 Lectures

Some examples of greedy algorithms. Terminology and implementation considerations. A look at some graph-related problems such as: finding a route (shortest paths); planning a communications network (minimum spanning trees); network routing management (flow graphs); compiling a program or planning a project (topological sorting).

Module Skills

Skills Type	Skills details
Application of Number	Yes, particularly in algorithm analysis.
Communication	Written skills will be needed to complete supporting documents to accompany assessed coursework.
Improving own Learning and Performance	See 2 above.
Information Technology	The whole module concerns this area.
Personal Development and Career planning	Carefully time management will be needed as so to enable students to complete coursework etc.
Problem solving	This is inherent in both the formative practical work and the assessed coursework.
Research skills	The students will need to search for and use relevant technical information while completing practical and assessed coursework.
Subject Specific Skills	Yes. See module title and content.
Team work	No.

Reading List

General Text

Alfred Aho, John Hopcroft, and Jeffrey Ullman (1983) *Data Structures and Algorithms* Addison-Wesley, Reading, Massachusetts [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+Structures+and+Algorithms+Alfred+Aho%2C+John+Hopcroft%2C+and+Jeffrey+Ullman&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+Structures+and+Algorithms+Alfred+Aho%2C+John+Hopcroft%2C+and+Jeffrey+Ullman&fn=search&vid=ABERU_VU1)

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (1995) *Design Patterns, Elements of Reusable Object-Oriented Software* Addison-Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Design+Patterns%2C+Elements+of+Reusable+Object-Oriented+Software+Erich+Gamma%2C+Richard+Helm%2C+Ralph+Johnson%2C+John+Vlissides&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Design+Patterns%2C+Elements+of+Reusable+Object-Oriented+Software+Erich+Gamma%2C+Richard+Helm%2C+Ralph+Johnson%2C+John+Vlissides&fn=search&vid=ABERU_VU1)

Freeman, Eric (2004.) *Head First design patterns /Eric Freeman, Elisabeth Freeman ; with Kathy Sierra and Burt Bates.* O'Reilly [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Head+First+design+patterns+%2FEric+Freeman%2C+Elisabeth+Freeman+%2B+with+Kathy+Sierra+and+Burt+Bates.+Freeman%2C+Eric&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Head+First+design+patterns+%2FEric+Freeman%2C+Elisabeth+Freeman+%2B+with+Kathy+Sierra+and+Burt+Bates.+Freeman%2C+Eric&fn=search&vid=ABERU_VU1)

- Knuth, Donald Ervin (1997.) *The art of computer programming /Donald E. Knuth*. 3rd ed. Addison-Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=The+art+of+computer+programming+%2FDonald+E.+Knuth.+Knuth%2C+Donald+Ervin&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=The+art+of+computer+programming+%2FDonald+E.+Knuth.+Knuth%2C+Donald+Ervin&fn=search&vid=ABERU_VU1)
- Knuth, Donald Ervin (1998.) *The art of computer programming /Donald E. Knuth*. 3rd ed. Addison-Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=The+art+of+computer+programming+%2FDonald+E.+Knuth.+Knuth%2C+Donald+Ervin&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=The+art+of+computer+programming+%2FDonald+E.+Knuth.+Knuth%2C+Donald+Ervin&fn=search&vid=ABERU_VU1)
- Knuth, Donald Ervin (1998.) *The art of computer programming /Donald E. Knuth*. 3rd ed. Addison-Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=The+art+of+computer+programming+%2FDonald+E.+Knuth.+Knuth%2C+Donald+Ervin&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=The+art+of+computer+programming+%2FDonald+E.+Knuth.+Knuth%2C+Donald+Ervin&fn=search&vid=ABERU_VU1)
- Martin Fowler (2003) *UML Distilled: A Brief Guide to the Standard Object Modeling Language (Addison-Wesley Object Technology S.)*, Addison Wesley. [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=UML+Distilled%3A+A+Brief+Guide+to+the+Standard+Object+Modeling+Language+%28Addison-Wesley+Object+Technology+S.%29%2C+Martin+Fowler&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=UML+Distilled%3A+A+Brief+Guide+to+the+Standard+Object+Modeling+Language+%28Addison-Wesley+Object+Technology+S.%29%2C+Martin+Fowler&fn=search&vid=ABERU_VU1)
- Michael Main (1999) *Data Structures and Other Objects, Using Java* Addison Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+Structures+and+Other+Objects%2C+Using+Java+Michael+Main&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+Structures+and+Other+Objects%2C+Using+Java+Michael+Main&fn=search&vid=ABERU_VU1)
- Rebecca Wirfs-Brock, Brian Wilkerson, and Lauren Wiener (1990) *Designing Object-Oriented Software* Prentice Hall [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Designing+Object-Oriented+Software+Rebecca+Wirfs-Brock%2C+Brian+Wilkerson%2C+and+Lauren+Wiener&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Designing+Object-Oriented+Software+Rebecca+Wirfs-Brock%2C+Brian+Wilkerson%2C+and+Lauren+Wiener&fn=search&vid=ABERU_VU1)
- T. Budd (2001) *Classic Data Structures in Java* Addison-Wesley Pub Co [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Classic+Data+Structures+in+Java+T.+Budd&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Classic+Data+Structures+in+Java+T.+Budd&fn=search&vid=ABERU_VU1)
- T.A. Standish (1998) *Data Structures in Java* Addison Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+Structures+in+Java+T.A.+Standish&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+Structures+in+Java+T.A.+Standish&fn=search&vid=ABERU_VU1)
- Weiss, Mark Allen. (1999.) *Data structures & algorithm analysis in Java /Mark Allen Weiss*. Addison-Wesley [Primo search \(http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+structures+and+algorithm+analysis+in+Java+%2FMark+Allen+Weiss.+Weiss%2C+Mark+Allen.&fn=search&vid=ABERU_VU1\)](http://primo.aber.ac.uk/primo_library/libweb/action/search.do?v1%28freeText0%29=Data+structures+and+algorithm+analysis+in+Java+%2FMark+Allen+Weiss.+Weiss%2C+Mark+Allen.&fn=search&vid=ABERU_VU1)

Notes

This module is at [CQFW \(http://wales.gov.uk/topics/educationandskills/qualificationsinwales/creditqualificationsframework/?lang=en\)](http://wales.gov.uk/topics/educationandskills/qualificationsinwales/creditqualificationsframework/?lang=en) Level 5