London South Bank University

Module Guide

Data Structures and Algorithms

CSI-4-DSA

https://my.lsbu.ac.uk

School of Engineering

Level 4

Table of Contents

1. MODULE DETAILS

Module Title:	Data Structure and Algorithms
Module Level:	4
Module Reference Number:	CSI-4-DSA
Credit Value:	20
Student Study Hours:	200
Contact Hours:	65
Private Study Hours:	135
Pre-requisite Learning (If applicable):	none
Co-requisite Modules (If applicable):	none
Course(s):	BSc (Hons) Computer Science
Year and Semester	Year 1 Semester 1
Module Coordinator:	Mike Child
MC Contact Details (Tel, Email, Room)	020-7815-7483, childm@lsbu.ac.uk, FW112
Teaching Team & Contact Details	Lucia Otoyo
(If applicable):	020-7815-7419, lucia.otoyo@lsbu.ac.uk, FW-218
Subject Area:	Software Development
Summary of Assessment Method:	Coursework
External Examiner appointed for module:	TBC

2. SHORT DESCRIPTION

This module teaches the definition of data structures, attributes, arrays, records, linked lists, binary trees and hash tables, using the fundamental elements of programming languages to construct them (for example using pointers). Also the derivation of algorithms, problem solving techniques, sequences, selections, and repetitions, sorting and searching. It covers pseudo-code, UML diagrams and how they can be used when representing algorithms, iterative and recursive algorithms and abstract data types. The relationship between abstract data types and object-oriented classes is introduced and the ready-made implementation of common structures such as hash tables in software libraries is explored.

3. AIMS OF THE MODULE

To build upon the fundamental knowledge acquired in the previous software development module to develop problem analysis and solving skills.

4. LEARNING OUTCOMES

4.1 Knowledge and Understanding

On completion of the module you will be able to:

- Describe different algorithms and the means used to measure their performance.

4.2 Intellectual Skills

On completion of the module you will be able to:

- Analyse programming problems and identify appropriate algorithmic solutions.

4.3 Practical Skills

On completion of the module you will be able to:

- Develop software to solve relatively complex problems and assess alternative solutions.

4.4 Transferable Skills

On completion of the module you will be able to:

- Apply critical and analytic reasoning to tasks.

5. ASSESSMENT OF THE MODULE

100% coursework

- 40% of the mark to come from a mid-module in-class test. This will consist of short answer questions in a written test.
- 50% of the mark to come from an end of module in lab development task, supplemented by some test questions.
- 10% of the mark to come from an engagement mark based upon how many of practical programming lab exercises the students complete and upload to the VLE.

6. FEEDBACK

Students will receive ongoing feedback on their coursework during the lab sessions, and will normally receive written feedback on coursework submissions within 15 days.

7. INTRODUCTION TO STUDYING THE MODULE

7.1 Overview of the Main Content

This module emphasises learning through practical exercises and the development of software artefacts. Short lectures will be used to inform the laboratory activities and provide a forum for discussion of issues students have encountered in the practical work. The lab sessions will occupy the majority of the contact time and will involve much independent working. Students are required to keep a clear record of the work they have done and are encouraged to experiment and investigate beyond the basic material being taught.

7.2 Overview of Types of Classes

There will be a combination of lectures delivering technical, theoretical and contextual information, and lab sessions in which students will work through practical exercises.

7.3 Importance of Student Self-Managed Learning Time

Although much laboratory time will be dedicated to practical exercises, students will need to spend more time in independent study on this. In addition, the theoretical material delivered in lectures will require independent reading and study in order to grasp it fully and prepare for the examination.

7.4 Employability

Software development skills are in demand in themselves, and understanding software development is an important skill in many non-development employment roles as it allows critical appraisal of solutions offered by others.

8. THE PROGRAMME OF TEACHING, LEARNING AND ASSESSMENT

The following outline is only indicative, the order and exact content of the lectures may vary according to unavoidable factors.

Week	Content
1	Introduction to module and development environments
2	Aggregate data types
3	Lists
4	Stacks and Queues
5	Searching
6	Hash tables
7	Maps and Sets
8	Revision/review and First assessment
9	Graphs and trees
10	Binary Search Trees
11	AVL balanced trees
12	
13	Revision/review and Second assessment

9. STUDENT EVALUATION

This module is being for the first time and therefore no previous student evaluation is available.

10. LEARNING RESOURCES

10.1 Core Materials

- Goodrich, M.T., Tamassia, R. (2014) *Datastructures and Algorithms in Java*. John Wiley & Sons; ISBN 1118808576
- Weiss, M. (2013) *Data Structures and Algorithm Analysis in C++*. Pearson; ISBN 0273769383
- Cormen, T. *et al*. (2009) *Introduction to Algorithms*. MIT Press; ISBN 0262533057

10.2 Optional Materials

NOTES

none