

CS1D1-30 Programming and Data Structures

23/24

Department

Computer Science

Level

Undergraduate Level 1

Module leader

Khalil Challita

Credit value

30

Module duration

8 weeks

Assessment

50% coursework, 50% exam

Study location

University of Warwick main campus, Coventry

Description

Introductory description

You cannot register for this module unless you are enrolled on the BSc Computer Science and Technology Solutions Degree Apprenticeship. It is not possible to request this module as an unusual option. If you are studying at Warwick as a visiting student from overseas it is not possible to register for this module.

The module will allow students to be conversant in the fundamentals of programming, from basic data types to language constructs. More advanced material will focus on features such as inheritance, polymorphism, and abstract classes. Students will also observe algorithms and testing alongside the main outcomes.

Module aims

The principle aims of this module are to introduce students to programming in both procedural and object-oriented styles using a high-level programming language. The module will also cover a selection of data structures and their associated properties. This module is a first course in computer programming.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

This module will give students an introduction to:

- Programming and the von Neumann paradigm
- Basics of programming syntax and specification
- Primitive data types
- Control structures and recursion
- Methods, scopes, and parameter passing
- Arrays
- Object-oriented programming
- Data encapsulation
- Abstract classes, interfaces, and generic typing
- Testing
- Abstract Data Types (linked lists, stacks, queues, graphs)
- Search and Sort algorithms

Learning outcomes

By the end of the module, students should be able to:

- Understand the programming process, from the definition of a problem and the design of a solution at an abstract level to an implementation of a solution.
- Develop programs in a high level programming language with the imperative paradigm.
- Develop complex programs in an object-oriented manner, demonstrating principles of encapsulation and abstraction.
- Demonstrate some knowledge of testing programs to ensure that they meet the defined specification.
- Implement common abstract data types from basic elements following good design principles.
- Implement search and sort algorithms efficiently, understanding the trade off between time and space complexity.
- Demonstrate an understanding of the limitations of abstract data types (ADT), and be able to select the most appropriate ADT for the task at hand.

Indicative reading list

Liang, D., "Introduction to Java Programming (11/e)", Pearson NY, 2019

Goodrich, MT, Tamassia, R., and Goldwasser, MH, "Data Structures and Algorithms in Java (6/e)", Wiley, 2015

Subject specific skills

- Design, implement, test and debug software to meet requirements using contemporary

methods including agile development

- Apply industry standard processes, methods, techniques and tools to execute projects
- Contemporary techniques for design, developing, testing, correcting, deploying and documenting software systems from specifications, using agreed standards and tools
- The fundamentals of data structures, database system design, implementation and maintenance

Transferable skills

- Applies analytical and critical thinking skills to Technology Solutions development and to systematically analyse and apply structured problem solving techniques to complex systems and situations
 - Have demonstrated that they have mastered basic business disciplines, ethics and courtesies, demonstrating timeliness and focus when faced with distractions and the ability to complete tasks to a deadline with high quality.
 - Flexible attitude
 - Ability to perform under pressure
 - A thorough approach to work
 - Logical thinking and creative approach to problem solving
-

Study

Study time

Type	Required
Lectures	40 sessions of 1 hour (13%)
Tutorials	14 sessions of 1 hour (5%)
Practical classes	35 sessions of 1 hour (12%)
Work-based learning	100 sessions of 1 hour (33%)
Other activity	111 hours (37%)
Total	300 hours

Private study description

No private study requirements defined for this module.

Other activity description

Self directed learning, assignments, revision

Costs

No further costs have been identified for this module.

Assessment

You do not need to pass all assessment components to pass the module.

Assessment group C2

	Weighting	Study time
Reflective essay on practice in area within workplace	10%	
Practical technical assessment	40%	
CS1D1 Programming and Data Structures	50%	

Feedback on assessment

Written feedback will be provided for the practical assignment and the essay

[Past exam papers for CS1D1](#)

Availability

Courses

This module is Core for:

- Year 1 of DCSA-I112 Undergraduate Computer Science and Technology Solutions (Data Analyst) (Degree Apprenticeship)