# Data Structures and Algorithms 1 (A13489)

**Short Title:** Data Structures & Algorithms 1
**Department:** Computing and Mathematics
**Credits:** 5 **Level:** Introductory

## Description of Module / Aims

Implement, from first principles, custom and general purpose data structures and algorithms that are efficient, thread safe, and robustly tested and validated.

## Programmes

| | stage/semester/status |
|---|---|
| BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B) | 2 / 3 / M |
| BSc in Software Systems Development (WD_KCOMC_D) | 2 / 3 / M |
| BSc (Hons) in Applied Computing (WD_KCOMP_B) | 2 / 3 / M |
| BSc (Hons) in Entertainment Systems (WD_KENTS_B) | 2 / 3 / M |
| BSc (Hons) in the Internet of Things (WD_KINTT_B) | 2 / 3 / M |
| BSc in Computing (ACCS) (WD_SR16IT_1) | 2 / 4 / M |
| BSc (Hons) in Software Systems Development (WD_SR16SD_1) | 2 / 3 / M |
| BSc (Hons) in Software Systems Development (International) (WD_SR16SD_2) | 2 / 3 / M |

## Indicative Content

- Implementing Data Structures: Abstract Data Types (ADTs); Lists; Sets; Stacks; Queues; Maps
- Implementing Search Algorithms: Linear; Binary; Hashing; Other
- Implementing Sorting Algorithms: Selection; Bubble; Insertion; Other
- Recursive Algorithms
- Multithreading and Concurrency Issues
- Test-Driven Development for Data Structures and Algorithms

## Learning Outcomes

*On successful completion of this module, a student will be able to:*

1. Construct, from first principles, custom and general purpose data structures.
2. Construct suitable and efficient search algorithms for different data structures.
3. Construct efficient sorting algorithms for different data structures.
4. Use recursion in algorithmic implementations.
5. Explain the issues surrounding, and be able to implement solutions for, concurrently accessed data structures.
6. Construct robust data structures and efficient algorithms in a systematic, test-driven fashion.

## Learning and Teaching Methods

- This module will be presented by a combination of lectures and computer-based practicals whilst capitalising on a web-enhanced learning environment.
- Students will work on building a portfolio of practical project work.
- Cooperative and peer learning (e.g. pair-programming, teamwork).
- Self-directed learning.

## Assessment Methods

| | Weighting | Outcomes Assessed |
|---|---|---|
| Continuous Assessment | 100% | |
| In-Class Assessment | 30% | 1,2,3,4,5 |
| Portfolio | 70% | 1,2,3,6 |

## Assessment Criteria

*<40%:* Unable to describe or implement basic data structures and algorithms.

*40%–49%:* Able to describe and implement basic data structures and algorithms.

*50%–59%:* Able to choose and apply appropriate data structures and algorithms to solve problems.

*60%–69%:* Able to choose and apply appropriate data structures and algorithms to solve complex problems.

*70%–100%:* Able to critically choose, analyse and design custom data structures and algorithmic solutions to a high standard for a range of both complex and unforeseen problems.

## Learning Modes

| Learning Type | F/T Hours | P/T Hours |
|---|---|---|
| Independent Learning | 75 | 111 |
| Practical | 36 | 12 |
| Lecture | 24 | 12 |

## Supplementary Material(s)

- "Khan Academy Computer science alogrithms." https://www.khanacademy.org/computing/computer-science/algorithms
- Sedgewick, R. and K. Wayne. *Algorithms*. NY: Addison-Wesley, 2011.

## Requested Resources

- Room Type: Computer Lab