# CS2D2-15 Programming paradigms

## 24/25

**Department**
> Computer Science

**Level**
> Undergraduate Level 2

**Module leader**
> Alexander Dixon

**Credit value**
> 15

**Module duration**
> 5 weeks

**Assessment**
> 100% coursework

**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.

This module will help students gain a familiarity with the concepts of main programming paradigms, including the basic principles underlying different programming paradigms, evaluation criteria and language implementation issues, and the strength and weaknesses of different paradigms. They will then apply the knowledge of several different paradigms to practical program development, in order to develop effective solutions to problems in the workplace.

### Module aims

This module introduces students to a range of different programming paradigms. It presents the concepts that underpin different approaches to programming and to the specific languages that implement them. It aims to provide students with the knowledge and skills needed to critically compare and evaluate different paradigms, to make informed choices about programming languages, and to develop efficient and effective programs.

## 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.

In this module, students will learn about:

- The principles of programming paradigms
- Comparative assessment of paradigms and implementation issues
- Understanding of evaluation criteria and language design issues
- Practical examples in a range of different paradigms such as imperative, declarative object oriented, logic, functional, multi-paradigm
- The strengths and weaknesses of different paradigms
- Applying the knowledge of several different paradigms to practical program development
- Developing effective solutions to problems using the most appropriate paradigm(s)
- Relating their understanding of different programming approaches to problem-solving in the workplace

## Learning outcomes

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

- Discuss and contrast the issues, features, design, and concepts of a range of programming paradigms and languages.
- Apply, through a suitable choice, a programming paradigm and language to solve a given problem.
- Demonstrate practical knowledge of languages representing several different paradigms.
- Apply knowledge of different programming approaches to problem-solving in the workplace.

## Indicative reading list

Toal, R., et al., "Programming Language Explorations", Chapman and Hall/CRC (2016)
Tate, BA, "Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages", Pragmatic Bookshelf (2010)
Scott, ML, "Programming Language Pragmatics (3/e)", Morgan Kaufmann publishers (2009)
Gabbrielli, M., and Martini, S., "Programming Languages: Principles and Paradigms", Springer (2010)

## Subject specific skills

- Knowledge of various programming paradigms and how to apply them

## Transferable skills

- 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 | 15 sessions of 1 hour (10%) |
| Tutorials | 14 sessions of 1 hour (9%) |
| Practical classes | 9 sessions of 2 hours 30 minutes (15%) |
| Work-based learning | 157 sessions of 30 minutes (52%) |
| Other activity | 20 hours (13%) |
| Total | 150 hours |

### Private study description

No private study requirements defined for this module.

### Other activity description

Self-directed learning

# 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 A

| | Weighting | Study time |
|---|---|---|
| Reflective report based on workplace assignment | 50% | |
| Discussion on programming principles | 50% | |

### Feedback on assessment

Written and verbal

---

## Availability

## Courses

This module is Core for:

- Year 2 of DCSA-I1I2 Undergraduate Computer Science and Technology Solutions (Data Analyst) (Degree Apprenticeship)