

# Unit 1: Programming

**Unit code** H/618/7388

**Unit type** Core

**Unit level** 4

**Credit value** 15

---

## Introduction

Programming involves describing processes and procedures that are derived from algorithms. The ability to program is what sets apart a developer and an end user. Typically, the role of the developer is to instruct a device (such as a computer) to carry out instructions; the instructions are known as source code and are written in a language that is converted into something the device can understand. The device executes the instructions it is given.

Algorithms help to describe the solution to a problem or task by identifying the data and the process needed to represent the problem or task *and* the set of steps needed to produce the desired result. Programming languages typically provide the representation of both the data and the process; they provide control constructs and data types (which can be numbers, words and objects, and be constant or variable). The control constructs are used to represent the steps of an algorithm in a convenient yet unambiguous fashion. Algorithms require constructs that can perform sequential processing, selection for decision making and iteration for repetitive control. Any programming language that provides these basic features can be used for algorithm representation.

This unit introduces students to the core concepts of programming along with an introduction to algorithms and the characteristics of programming paradigms. Among the topics included in this unit are: introduction to algorithms, procedural, object-orientated and event-driven programming, security considerations, the integrated development environment and the debugging process.

On successful completion of this unit, students will be able to design and implement algorithms in a chosen language in a suitable Integrated Development Environment (IDE). This IDE will be used to develop and help track any issues with the code. As a result, students will develop skills such as communication literacy, critical thinking, analysis, reasoning and interpretation, which are crucial for gaining employment and developing academic competence.

## **Learning Outcomes**

By the end of this unit students will be able to:

- LO1 Define basic algorithms to carry out an operation and outline the process of programming an application
- LO2 Explain the characteristics of procedural, object-orientated and event-driven programming
- LO3 Implement basic algorithms in code using an IDE
- LO4 Determine the debugging process and explain the importance of a coding standard.

## Essential Content

### LO1 Define basic algorithms to carry out an operation and outline the process of programming an application

#### *Algorithm definition:*

Writing algorithms to carry out an operation, e.g. Bubble sort.

The relationship between algorithms and code.

The generation process of code; the roles of the pre-processor, compiler and linker, interpreter.

### LO2 Explain the characteristics of procedural, object-orientated and event-driven programming

#### *Characteristics of code:*

Definitions of: data types (the role of constants/variables), data structures, e.g. arrays, stacks, queues, methods (including input/output), control structures, iteration, scope, parameter passing, classes, inheritance and events.

Key components of an IDE, with a brief explanation of each component.

Use of addition of advanced text editors to view code, such as Notepad++, Atom, Sublime Text etc.

### LO3 Implement basic algorithms in code using an IDE

#### *Implementation:*

Develop simple applications that implement basic algorithms, including the features of a suitable language and IDE.

Create logical and maintainable codes.

Consideration of security concerns and how they could be solved.

Build, manage and deploy code to the relevant environment to solve the identified problems.

**LO4 Determine the debugging process and explain the importance of a coding standard**

*Review and reflection:*

Documentation of the debugging process in the IDE, with reference to watch lists, breakpoints and tracing.

Use of debugging the process to help developers fix vulnerabilities, defects and bugs in code.

Apply structured techniques to problem solving, debugging code and consider structure of programmes to identify and resolve issues.

Understand coding standards and their benefits when writing code.

## Learning Outcomes and Assessment Criteria

Pass	Merit	Distinction
<b>LO1</b> Define basic algorithms to carry out an operation and outline the process of programming an application		<b>D1</b> Evaluate the implementation of an algorithm in a suitable language and the relationship between the written algorithm and the code variant.
<b>P1</b> Define an algorithm and outline the process in building an application. <b>P2</b> Determine the steps taken from writing code to execution.	<b>M1</b> Analyse the process of writing code, including the potential challenges faced.	
<b>LO2</b> Explain the characteristics of procedural, object-orientated and event-driven programming		<b>D2</b> Critically evaluate the source code of an application that implements the procedural, object-orientated and event-driven paradigms, in terms of the code structure and characteristics.
<b>P3</b> Discuss what procedural, object-orientated and event-driven paradigms are; their characteristics and the relationship between them.	<b>M2</b> Compare the procedural, object-orientated and event-driven paradigms used in given source code of an application.	
<b>LO3</b> Implement basic algorithms in code using an IDE		<b>D3</b> Evaluate the use of an IDE for development of applications contrasted with not using an IDE.
<b>P4</b> Write a program that implements an algorithm using an IDE.	<b>M3</b> Enhance the algorithm written, using the features of the IDE to manage the development process.	
<b>LO4</b> Determine the debugging process and explain the importance of a coding standard		<b>D4</b> Evaluate the role and purpose of a coding standard and why it is necessary in a team as well as for the individual.
<b>P5</b> Explain the debugging process and the debugging facilities available in the IDE. <b>P6</b> Explain the coding standard you have used in your code.	<b>M4</b> Examine how the debugging process can be used to help develop more secure, robust applications.	

## Recommended Resources

This unit does not specify which programme language should be used to deliver the content – this decision can be made by the tutor.

Examples of languages that are used in industry are C#, Python, Ruby and Java, but any language that will allow students to achieve the Learning Outcomes is acceptable.

### Textbooks

Aho, A. V. et al. (1987) *Data Structures and Algorithms*. 1st Ed. Addison-Wesley.

Hunt, A. et al. (2000) *The Pragmatic Programmer: From Journeyman to Master*.

1st Ed. Addison-Wesley.

McConnell, S. (2004) *Code Complete: A Practical Handbook of Software Construction*. 2nd Ed. Microsoft Press.

### Links

This unit links to the following related units:

*Unit 19: Data Structures & Algorithms*

*Unit 20: Applied Programming and Design Principles*

*Unit 54: Prototyping.*