

## Module Overview – Computing

The Computing 1.0 module sets out essential concepts and skills relating to the use of computational thinking and coding to create simple computer programs.

Many job roles that are not explicitly focused on software development required an understanding of the principles of computational thinking and the foundations of coding. The Computing module covers core computational thinking techniques like problem decomposition, pattern recognition, abstraction and algorithms as well as practical skills relating to coding.

## Module Outline

Category	Skill set
Computing terms	<ul style="list-style-type: none"> <li>Key concepts</li> </ul>
Computational thinking methods	<ul style="list-style-type: none"> <li>Problem analysis</li> <li>Algorithms</li> </ul>
Starting to code	<ul style="list-style-type: none"> <li>Getting started</li> <li>Variables and data types</li> </ul>
Building using code	<ul style="list-style-type: none"> <li>Logic</li> <li>Iteration</li> <li>Conditionality</li> </ul>

	<ul style="list-style-type: none"> <li>• Procedures and functions</li> <li>• Events and commands</li> </ul>
Test, debug and release	<ul style="list-style-type: none"> <li>• Run, test and debug</li> <li>• Release</li> </ul>

## What are the benefits of this module?

- Covers the key skills and main concepts relating to computational thinking and coding.
- Certifies best practice in computational thinking and coding.
- Introduces concepts and skills that are essential for anyone interested in developing specialised IT skills.
- Assists in developing generic problem solving skills that are useful for everyone.
- Developed with input from subject matter experts and practising computing professionals from around the world. This process ensures the relevance and range of module content.

## Learning outcomes

On completion of this module the candidate will be able to:

- Understand key concepts relating to computing and the typical activities involved in creating a program.

## AdvancedSkills@Work

---

- Understand and use computational thinking techniques like problem decomposition, pattern recognition, abstraction and algorithms to analyse a problem and develop solutions
- Write, test and modify algorithms for a program using flowcharts and pseudocode.
- Understand key principles and terms associated with coding and the importance of well-structured and documented code.
- Understand and use programming constructs like variables, data types, and logic in a program.
- Improve efficiency and functionality by using iteration, conditional statements, procedures and functions, as well as events and commands in a program.
- Test and debug a program and ensure it meets requirements before release.

### Training Duration

16 hours

### Medium of Instruction

Instructor Led Class

### Certification

Successful trainees will receive the internationally recognised ICDL Certificate upon completion of the course.