

2024 / 25

School of Science and Computing

☎ +353 (0)51 302037

✉ [Eleanor.Reade@setu.ie](mailto:Eleanor.Reade@setu.ie)

🌐 [www.wit.ie/schools/science\\_computing](http://www.wit.ie/schools/science_computing)



**SE  
TU**

Ollscoil  
Teicneolaíochta  
an Oirdheiscirt

South East  
Technological  
University

## Module Descriptor

---

### Programming Fundamentals - ICT (Computing and Mathematics)

# Programming Fundamentals - ICT (A14167)

**Short Title:** Programming Fundamentals - ICT  
**Department:** Computing and Mathematics  
**Credits:** 10

**Level:** Introductory

## Description of Module / Aims

This module will introduce the student to fundamental programming techniques. The student will learn how to design algorithms using pseudocode, write small applications based on their designs and test their written applications. It assumes no prior knowledge of technology, programming languages or programming environments. Using basic techniques, the student should gain confidence in producing small applications. The module will also further develop the thought processes, problem solving techniques and programming structures essential for developing larger systems responsible for more complex tasks. The module will focus on developing maintainable, robust, persistent applications that are consistent with user experience (UX) best practice.

## Programmes

stage/semester/status

COMP-0516 Higher Diploma in Science in Computer Science (WD\_KCOSC\_G)

1 / 1 / M

## Indicative Content

- Problem solving approaches
- Control structures for programming
- Programming language constructs
- UX approaches using data-entry validation and error recovery
- Hands-on debugging, testing and documentation activities
- Use of innovative visualisation tools, IDEs and frameworks
- Introduction to industry accepted standards specific to the programming language
- Use of complex constructs and data structures associated with the chosen language
- Introduction to UX
- Programming for persistence
- Programming techniques for well-behaved, robust applications
- Read, understand and consume API specific to the chosen language
- Use of innovative visualisation tools, IDEs and frameworks

## Learning Outcomes

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

1. Apply core problem solving approaches suitable to the programming discipline to build algorithms.
2. Construct small applications using standard sequence, conditional and iterative control structures.
3. Change and expand small applications.
4. Construct small applications that use simple UI, computation and data structures.
5. Apply techniques to effectively test, debug and document small applications.
6. Defend and explain how the above applications work.
7. Apply problem-solving strategies to various computing problems of increasing complexity.
8. Plan, code, test and document applications using advanced programming constructs and data structures.
9. Construct applications consistent with UX best practice.
10. Construct persistent applications.
11. Apply maintainability and robustness when designing applications.

## Learning and Teaching Methods

- Combination of lectures and computer-based practical labs.
- Cooperative learning/peer tutoring (i.e. pair-programming for some practical labs, Problem-based learning approaches for some assignments).
- Self-directed learning.

## Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	48	
Practical	48	
Independent Learning	174	

## Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Assignment	30%	1,2,3
Assignment	35%	3,4,5,6
Assignment	35%	5,6,7,8,9,10,11

## Assessment Criteria

- <40%: Inability to design, develop and test maintainable, persistent, robust UX applications to solve a particular problem.
- 40%–49%: Ability to design, develop and test maintainable, persistent, robust UX applications to solve a straight-forward problem.
- 50%–59%: Comfortable with designing, developing and testing maintainable, persistent, robust UX applications to solve problems similar to those presented in the module.
- 60%–69%: Proficient with designing, developing and testing maintainable, persistent, robust, high-quality UX applications to solve complex problems.
- 70%–100%: Proficient with designing, developing and testing maintainable, persistent, robust, high-quality, elegant UX applications to solve complex problems that are substantially different to those studied in the module.

## Supplementary Material(s)

- Kolling, M. *Objects first with Java-A Practical Introduction using BlueJ. 5th ed...* New York: Pearson Education, 2012.
- Sprankle, M. *Problem Solving and Programming Concepts*. New York: Prentice Hall, 2011.

## Requested Resources

- Computer Lab: BYOD Lab