2024 / 25

School of Science and Computing
☎ +353 (0)51 302037
✉ Eleanor.Reade@setu.ie
www.wit.ie/schools/science_computing

SETU
Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Module Descriptor

Programming Fundamentals 1 (Computing and Mathematics)

Programming Fundamentals 1 (A06609)

Short Title:	Programming Fundamentals 1		
Department:	Computing and Mathematics		
Credits:	5	**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.

Programmes

		stage/semester/status
COMP-0008	BEng (Hons) in Electrical and Automation Engineering (International) (WD_ETRIC_BI)	1 / 1 / M
COMP-0008	BEng (Hons) in Information Engineering (International) (WD_EEELC_BI)	1 / 1 / M
COMP-0008	BSc (Hons) in Applied Computing (International) (WD_KACCM_BI)	1 / 1 / M
COMP-0008	BSc (Hons) in Applied Computing (WD_KACCM_B)	1 / 1 / M
COMP-0008	BSc (Hons) in Applied Computing (WD_KCOMP_B)	1 / 1 / M
COMP-0008	BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B)	1 / 1 / M
COMP-0008	BSc (Hons) in Computer Science (WD_KCMSC_B)	1 / 1 / M
COMP-0008	BSc (Hons) in Physics for Modern Technology (WD_KPHTE_B)	2 / 3 / M
COMP-0008	BSc (Hons) in Software Engineering (WD_KDEVP_BI)	1 / 1 / M
COMP-0008	BSc (Hons) in Software Systems Development (WD_KDEVP_B)	1 / 1 / M
COMP-0008	BSc (Hons) in the Internet of Things (International) (WD_KINTT_BI)	1 / 1 / M
COMP-0008	BSc in Applied Computing (WD_KCOMP_D)	1 / 2 / M
COMP-0008	BSc in Information Technology (WD_KINFT_D)	1 / 2 / M
COMP-0008	BSc in Software Systems Development (WD_KCOMC_D)	1 / 1 / M
COMP-0008	Diploma in Computing with Security and Forensics (WD_BCSEC_SP)	3 / 1 / M

Indicative Content

- Problem solving approaches
- Control structures for programming: sequence; selection; iteration
- Programming language constructs: variables; operators; methods; arrays
- Classes and objects
- User interaction (UI) approaches using data-entry validation and error recovery
- Hands-on debugging, testing and documentation activities
- Use of innovative visualisation tools, Integrated Development Environments (IDEs) and frameworks
- Introduction to industry accepted standards specific to the programming language

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. Code small applications using standard sequence, conditional and iterative control structures.
3. Change and expand small applications.
4. Code 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.

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.
- The lectures will be used to introduce new topics and their related concepts.
- A cooperative learning/peer tutoring (i.e. pair-programming for some practical labs) approach will be adopted during the practical sessions.
- Self-directed learning will be encouraged throughout the duration of the module.

Learning Modes

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

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Assignment	30%	1,2,3,6
Assignment	70%	1,2,4,5,6

Assessment Criteria

<40%: Inability to write, run test and debug small applications. Inability to understand, explain the workings of or modify, a small application. Inability to produce an algorithm to solve a simple problem.

40%–49%: Able to write, run, test and debug small applications comfortably. Able to understand, explain the workings of or modify a small application. Can produce an algorithm to solve a simple problem.

50%–59%: All the above and in addition can correctly choose appropriate programming constructs and data structures. Shows a high level of competency when writing algorithms.

60%–69%: All the above and in addition, be able to apply given solutions to new, similar problems. Starts to use Application Programming Interfaces (APIs) to research new capabilities. Starts to see applications in terms of systems of components.

70%–100%: All previous to an excellent level. Starts to understand the concept of effective solutions.

Essential Material(s)

- "Code Academy." http://www.codecademy.com/
- "Khan Academy." https://www.khanacademy.org/
- "W3 Schools on-line web tutorials." https://www.w3schools.com/

Supplementary Material(s)

- "BlueJ." http://www.bluej.org
- "Project Euler." https://projecteuler.net/
- "Python Tutorials." http://www.introtopython.org/
- Kolling, M. and D. Barnes. *Objects first with Java - A Practical Introduction using BlueJ.* 5th ed.. NY: Pearson, 2012.
- Sprankle, M. *Problem Solving and Programming Concepts.* 9th Ed.. NY: Prentice Hall, 2011.
- Venit, S. and E. Drake. *Prelude to Programming: Concepts & Design.* NY: Pearson Higher Education, 2015.
- Vickers, P. *How to think like a programmer: problem solving for the bewildered.* NY: Cengage, 2008.

Requested Resources

- Room Type: Computer Lab