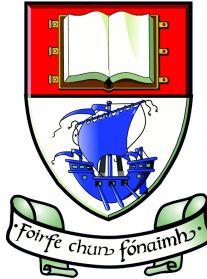


Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



School of Science and Computing

**Department of
Computing and Mathematics**

Module Descriptor

**Programming Fundamentals 2
(Computing and Mathematics)**

Programming Fundamentals 2 (A03902)

Short Title: Programming Fundamentals 2
Department: Computing and Mathematics
Credits: 5

Level: Introductory

Description of Module / Aims

The purpose of this module is to 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 interaction (UI) best practice.

Programmes

stage/semester/status		
COMP-0378	BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B)	1 / 2 / M
COMP-0378	BSc in Software Systems Development (WD_KCOMC_D)	1 / 2 / M
COMP-0378	BSc (Hons) in Applied Computing (WD_KCOMP_B)	1 / 2 / M
COMP-0378	BSc (Hons) in Entertainment Systems (WD_KENTS_B)	1 / 2 / M
COMP-0378	BSc in Information Technology (WD_KINFT_D)	2 / 3 / M
COMP-0378	BSc (Hons) in the Internet of Things (WD_KINTT_B)	1 / 2 / M
COMP-0378	BSc (Hons) in Physics for Modern Technology (WD_KPHTE_B)	2 / 4 / M
COMP-0378	BSc in Computing (ACCS) (WD_SR16IT_1)	2 / 3 / M
COMP-0378	BSc (Hons) in Software Systems Development (WD_SR16SD_1)	1 / 2 / M
COMP-0378	BSc (Hons) in Software Systems Development (International) (WD_SR16SD_2)	1 / 2 / M

Indicative Content

- Use of complex constructs and data structures associated with the chosen language
- Introduction to UI
- Programming for persistence
- Programming techniques for well-behaved, robust applications
- Use of Test Driven Development (TDD) approaches
- Read, understand and consume the Application Programming Interface (API) specific to the chosen language
- Use of innovative visualisation tools, Integrated Development Environments (IDEs) and frameworks

Learning Outcomes

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

1. Apply problem-solving strategies to various computing problems of increasing complexity.
2. Plan, code, test and document applications using advanced programming constructs and data structures.
3. Construct applications consistent with UI best practice.
4. Construct persistent applications.
5. Apply maintainability and robustness when designing applications.

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, problem-based learning for one assignment) approach will be adopted during the practical sessions.
- Self-directed learning will be encouraged throughout the duration of the module.

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Assignment	50%	1,2,4,5
Assignment	50%	1,2,3,5

Assessment Criteria

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

Learning Modes

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

Essential Material(s)

- "Code Academy." <http://www.codecademy.com/>
- "Khan Academy." <https://www.khanacademy.org/>
- "W3 Schools on-line Web Tutorials." 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: Prentice Hall/Pearson Education, 2012.
- Sprankle, M. *Problem Solving and Programming Concepts*. 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