FP016-30 Computer Science

22/23

Department
Warwick Foundation Studies
Level
Foundation
Module leader
Zahid Khan
Credit value
30
Assessment
40% coursework, 60% exam
Study location
University of Warwick main campus, Coventry

Description

Introductory description

FP016-30 Computer Science

Module web page

Module aims

This module combines theory of computing with practical computing activities including programming and application of the software lifecycle.
It aims to introduce students to the fundamental aspects of the academic discipline of Computer Science, illustrating the use of formal languages in computer science, including algorithms and programming.
It aims to develop students' computing-related problem-solving skills whilst also enabling students to apply computing skills to other areas and provides a suitable preparation for higher education courses in computing and related areas.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

Part 1: Algorithms and Computational Thinking
Introduction to Computer Science

Computational Thinking
Algorithms for Searching and Sorting
Data Structures

Part 2: Python Programming
What is a computer program?
Python Programming Language
Variables and Data Types
Selection
Repetition
Functions & Recursion
Development Environments
Individual Python Programming Project (30%)

Part 3: Data Representation
Numbers and Number Systems
Data Representation and Transformation
Real Numbers

Part 4: Computer Systems and architecture
Computer Logic
Hardware and Components
Operating Systems
Communication and Networks
Machine Code
Boolean Mathematics
Boolean Circuit Design
Binary and Bit Manipulation
Data Integrity

Part 5: Ethics and Consequences
Data Security and Privacy
Ethics of Computing
Consequences of Computing Group Presentation (10%)

Part 6: Software Engineering and Business
Computing in Business and Society
Top-Down Design & Specification
Implementation & Testing
Deployment & Maintenance

Page 7: Object Orientated Paradigm
OOP Languages (python)
Classes
Encapsulation
Inheritance
Polymorphism

Learning outcomes

By the end of the module, students should be able to:

- Recognise and show understanding of specific computing terminology, technology, key developments, principles and methods;
- Apply computational thinking to translate a range of problems into a form that can be solved by the computer;
- Use a combination of practical programming skills and theoretical knowledge to solve calculations, manipulate data and predict outcomes;
- Show an appreciation of the social, legal and ethical consequences of computer applications and technology;
- Display professional software engineering skills such as team working, time management and the importance of commercial reality; and
- Present and disseminate computing concepts, solutions and results including the necessary evidence to a range of audiences.

Indicative reading list

Dale, N and Lewis, J. (2015) Computer Science Illuminated (6th ed.) Jones and Bartlett Burdett, A. et al. (2013) BCS Glossary of Computing and ICT (13th ed.) BCS, The Chartered Institute for IT Brookshear, G and Brylow, D (2014) Computer Science: An Overview (12th ed) Pearson Shaw, Z (2017) Learn Python 3 the Hard Way. Addison-Wesley
Python Crash Course. Eric Matthes(2019) Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition

Interdisciplinary

Interdisciplinary links with mathematics.

Subject specific skills

To develop students use of analysis to interpret and critique existing historiographical debates and draw their own conclusions from the examination of primary sources.

Transferable skills

No transferable skills defined for this module.

Study

Study time

Type	Required
Lectures	25 sessions of 1 hour (25%)
Seminars	25 sessions of 3 hours (75%)
Total	100 hours

Type	Required
External visits	1 session of (0%)
Total	100 hours

Private study description

Private Study, to be taken out on improving and practicing programming techniques in preparation for exam and programming project.

Costs

No further costs have been identified for this module.

Assessment

You must pass all assessment components to pass the module.

Assessment group D3

	Weighting	Study time
Project Report (2000 words)	30%	30 hours

Students create a piece of software using the Python programming language and then write a critical review of the Python program they have developed. Emphasis is on the software engineering process and the techniques and algorithms learned during the course.

	Weighting	Study time
Presentation Computer Ethics	10%	10 hours

Presentation such as 'presentation detailing an area of ethics and associated consequences of computer technology in computer science'.

	Weighting	Study time
Online Examination	60%	30 hours

2 hour examination (Summer)

~Platforms - AEP

- Online examination: No Answerbook required
- Students may use a calculator

Feedback on assessment

Weekly seminars and tutorials will provide regular opportunities for students to discuss and debate the key issues and explore possible solutions. These discussions will be moderated by the tutor and feedback will be given in the seminars. Progress will be monitored via formative assessments, and the associated feedback will be returned via Tabula. Summative assessments will be marked with both written and verbal feedback provided to students in order to further develop their skills.

Past exam papers for FP016

Availability

Courses

This module is Core for:

- FIOE Warwick International Foundation Programme
    - Year 1 of FP18 Warwick International Foundation Programme - Computer Science
    - Year 1 of FP19 Warwick International Foundation Programme - Engineering

This module is Core option list A for:

- FIOE Warwick International Foundation Programme
    - Year 1 of FP13 Warwick International Foundation Programme - Mathematics and Economics
    - Year 1 of FP12 Warwick International Foundation Programme - Science and Engineering