MODULE PROFORMA		
Full module title: Computer Systems Fundamentals		
Module code: 4COSC004W	Credit level: 4	Length: 1 Semester
UK credit value: 20	ECTS value: 10	
College and School: College of Design, Creative and Digital Industries, School of Computer Science and Engineering		
Module Leader(s):		
Extension:	Email:	
Host course and course leader: BSc Computer Science		
Status: Core - BSc Computer Science, BEng Software Engineering		
Subject Board: COMENG		
Pre-requisites: none	Co-requisites: none	
Study abroad: No		
Special features: none		
Access restrictions: none		
Are the module learning outcomes delivered, assessed or supported through an arrangement with an organisation(s) other than the University of Westminster. No		
Summary of module content: This module is centred on the fundamental aspects of the way that a typical computer function either as a standalone entity or as part of a computer network. Discussion of the main hardware components of a computer system provides the backdrop to introduce the Von Neumann fetch–decode–execute cycle as well as the way in which data and information are stored in the computer. Students will learn about the various number systems (denary, binary and hexadecimal) that are utilised in computer systems. Progressing from the hardware level, students are introduced to the lowest programming level that humans can understand in the form of assembly programming. Above the assembly layer, there is the operating system layer. Students will be able to apply their knowledge of Binary number conversions and Logical Operations to perform Networking Calculations.		

Learning outcomes

By the end of the module the successful student will be able to:

LO1 Confidently convert between number systems, and represent negative and real values in the manner in which such are stored in computers.

LO2 Demonstrate a thorough and critical understanding of the methodology employed by the Operating System in managing processes, memory and the file systems including multi–threading and concurrency.

LO3 Show a thorough understanding of the von Neumann architecture and stored program paradigm.

LO4 Demonstrate a comprehensive understanding of the various network topologies, models, protocols, interfaces, IP address configuration and network sub–netting.

LO5 Demonstrate a thorough knowledge of the line commands interface by using some simple examples.

Course outcomes the module contributes to:

BEng Software Engineering	L4.3
BSc Computer Science	L4.3

Indicative syllabus content

- Foundations of number systems and how they are used in computer storage, the arithmetic and logical operations on these numbers (e.g. addition, subtraction, AND, OR), the representations of these numbers (e.g. signed and unsigned integers, characters, negative integers, fractions and IEEE754), and the standard character codes (e.g. ASCII, UNICODE)
- Principles of the Von Neumann fetch-decode-execute cycle
- As an example: Fundamental understanding of Assembly Language, ability to predict what a piece of Assembly Language will do.
- Functions of an operating system (e.g. process, memory and file system management), and file management features of an operating system (e.g. directory structures, file protection, simple tools and utilities, file and directory handling, partition tables and file systems)
- Introduction to computer networks – models (e.g. OSI, TCP/IP), topologies (e.g. star, mesh), protocols (e.g. SMTP), IP addressing and sub–netting, cloud services
- Introduction to the system commands through Command Line Interface.

Teaching and learning methods

Typically, students attend a lecture (2 hours) and a practical tutorial (2 hours) each week over semester two. The lecture sessions are used to explain the fundamental principles and to demonstrate the practicalities associated with a typical computer system. Lectures will include physical lectures, live webinars and recorded videos.

Practical laboratory exercises and interactive web pages are provided to focus the student's attention on the development of skills and knowledge associated with this module.

Automated feedback will be provided to the practical exercises, these will take the form of Java applications provided by publishers and interactive Excel spreadsheets.

There is a significant element of self–directed learning and the tutorial sessions provide the necessary support for these. Exercises are made available on the VLE to support both the tutorials and self–directed learning activities. Students are expected to put in a greater amount of effort in independent self–directed study in between classes. Guidance is provided to focus the self–directed study. Students taking this module are expected to have 144 hours for the preparation for the assessment.

Activity type	Category	Student learning and teaching hours*
Lecture /Webcast lecture	Scheduled	24
Seminar	Scheduled	
Tutorial	Scheduled	24
Project supervisor	Scheduled	
Demonstration	Scheduled	
Practical Classes and workshops	Scheduled	8
Supervised time in studio/workshop	Scheduled	
Fieldwork	Scheduled	
External visits	Scheduled	
Work-based learning	Scheduled	
Total Scheduled		56
Placement	Placement	
Independent study	Independent	144
Total student learning and teaching hours		*200*

*the hours per activity type are indicative and subject to change.

Assessment rationale

The two in–class tests assess the students' ability to confidently convert between number systems and represent data in the form it will be stored in a computer (LO1), the functions and services provided by the operating system (LO2), the Von-Neumann model (LO3), computer networks (LO4) and shell programming (i.e. command line interface) (LO5). Students gain such knowledge and understanding by attendance at lectures and doing practical exercises in tutorials and self–directed learning, and research.

Formative tutorial exercises and sample formative tests will be made available on Blackboard for students to reinforce and practise their knowledge and learning and assess progress. These exercises will be made available on a weekly basis. Typically, several exercises will be available each week and students are encouraged to do them both during the timetabled tutorial classes as well as in their own time because the exercises will be available at all relevant times. These assessments will cover all Learning Outcomes.

Assessment criteria

Criteria for assessments are designed with reference to the University's generic criteria to measure students' ability to meet the learning outcomes of a module. Specifically, within this module you will find detailed grading descriptors as part of each assessment.

To pass the module, students must demonstrate a basic understanding of the structure and operation of a computer and the numerical theory at its basis. Higher levels of achievement require students to demonstrate greater levels of understanding and the ability to convey that understanding, and technical competence in tutorial exercises and in the achievements in the two in–class tests.

Assessment methods and weightings

Assessment name	Weighting %	Qualifying mark %	Qualifying set	Assessment type (e.g. essay, presentation, open exam or closed exam)
In–Class Test 1	*50*	*30*		*In-Class Test/Assignment exam conditions*
In–Class Test 2	*50*	*30*		*In-Class Test/Assignment exam conditions*

Synoptic assessment

There are no synoptic assessments in this module.

Sources

Link to the online reading list

2020/21 Reading List