LONDON CAPITAL COMPUTER COLLEGE

Advanced Diploma in Computer Science (907) – Computer Systems Architecture

Prerequisites: Good computing knowledge	**Corequisites:** A pass or better in Diploma in System Analysis & Design or equivalence.

Aim: The course covers how programs are represented and executed by modern computers, low-level machine representations of programs and data; an understanding of how computer components influence program performance, assembly level machine organisation, memory system organisation and architecture, functional organisation, multiprocessing and alternative architectures. Fundamental concepts of the architectural structure and organisation of computers are reviewed, including fundamental execution cycle, central processing unit, input/output unit and memory management unit. The course reviews key abstractions supported at the architectural level such as virtual memory, micro-architecture, I/O controllers and processors. An analysis of the evolution of the major architectures from Complex Instruction Set Computers (CISC) to Reduced Instruction Set Computers (RISC) is carried out. Conceptual development and implementation of data structures including arrays, records, linear lists, stacks, queues and binary trees. Operating system structures, concurrent processes, resource scheduling, memory management, file system and protection and distributed systems are analysed in detail.

Required Materials: Recommended learning resources.	**Supplementary Materials:** Lecture notes and tutor extra reading recommendations.

Special Requirements: A thorough understanding on computer organisation, operating systems and data structures is required to enable candidates pass the examination.

Intended Learning Outcomes:	**Assessment Criteria:**
1 Define a computer system. Understand system capabilities and limitations.	1.1 Describe Input-Process-Output-Storage model 1.2 Analyse computer architecture components 1.3 Describe hardware components – CPU, memory and software components 1.4 Describe communications components 1.4 Describe protocols, standards and history of computers.
2 Describe numbering system computation. Define bits, data types and operations.	2.1 Define why binary is important 2.2 Define decimal, binary, octal and hexadecimal systems 2.3 Demonstrate binary arithmetic (addition, subtraction and multiplication) 2.4 Demonstrate how to compute fractions. 2.5 Define how data is represented in a computer 2.6 Define ASCII characters
3 Define the different data formats. Describe specifications for converting data into computer usable form. Define the different ways human data may be represented, stored and processed by a computer.	3.1 Describe sources of data 3.2 Identify common data representation types 3.3 Define character and control codes 3.4 Define image data.
4 Describe how integer data is represented. Define value/magnitude and sign (plus or minus).	4.1 Define a 16, 32 and 64-bit word 4.2 Define unsigned numbers 4.3 Define sign and magnitude 4.4 Define data overflow
5 Define floating point numbers. Analyse how floating point numbers are used in computer	5.1 Describe the exponential notation 5.2 Illustrate overflow and underflow 5.3 Describe normalisation

when the number is outside the integer range of the computer or contains a decimal fraction.	
6 Define the components of the CPU. Describe the von Neuman Model.	6.1 Describe the fetch execute cycle 6.2 Define bus characteristics 6.3 Describe general registers 6.4 Describe special-purpose registers 6.5 Identify memory operations and the relationship between memory address registers, memory data register and memory 6.6 Describe memory capacity 6.7 Define Random Access Memory (RAM) 6.8 Define Read Only Memory (ROM) 6.9 Define Point-to-point vs multipoint 6.10 Describe the motherboard layout 6.11 Describe the instruction set format
7 Define data structures. Illustrate the purpose of data structures.	7.1 Describe linear lists; stacks; queues; arrays and binary trees 7.2 Illustrate the process of traversing data 7.3 Describe how to add and delete data 7.4 Describe how to sort data 7.5 Define the process of searching for a specific item of data
8 Describe Assembly as a low level language.	8.1 Describe how assembly language is compiled. 8.2 Define assembly language instruction format.
9 Describe the difference between CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer).	9.1 Describe the CISC architecture 9.2 Describe the limitations of CISC architecture 9.3 Define RISC features 9.3 Describe Very Long Instruction Word (VLIW) architecture 9.4 Describe EPIC (Explicitly Parallel Instruction Computer) architecture 9.5 Define how paging is managed by the operating system. 9.6 Differentiate logical vs physical addresses 9.7 Define cache memory. Describe the difference between cache and virtual memory
10 Define how the processing speed or program execution is determined primarily by the ability of Input/Output (I/O) operations to stay ahead of the processor.	10.1 Describe I/O speed and coordination issues 10.2 Describe I/O device interface issues 10.3 Describe Input/output module functions. 10.4 Define the CPU interrupts. Explain the use of interrupts. 10.5 Define Direct memory access (DMA) 10.5 Define data bus configuration architecture 10.6 Describe different external bus and port interfaces. 10.7 Explore the structure of an operating system's I/O subsystem 10.8 Discuss the principles of I/O hardware and its complexity 10.9 Provide details of the performance aspects of I/O hardware and software

11 Describe computer peripherals, their classifications and how they are connected. Describe characteristics and features of Real-Time systems.	11.1 Describe storage devices and their data access time 11.2 Describe the hard disk layout format. 11.3 Describe the CD-ROM layout. 11.4 Explain the timing requirements of real-time systems 11.5 Distinguish between hard and soft real-time systems 11.6 Discuss the defining characteristics of real-time systems 11.7 Describe scheduling algorithms for hard real-time systems
12 Describe the difference in layout between PC and mainframe systems. Define clustering, mass-storage systems and distributed system structures. Describe Distributed File System (DFS) implementation overview.	12.1 Explore PC and mainframe components 12.2 Describe multiprocessing symmetrical processing 12.3 Describe cluster models 12.4 Describe the client-server architecture 12.5 Define parallel computing 12.6 Describe the physical structure of secondary and tertiary storage devices and the resulting effects on the uses of the devices 12.7 Explain the performance characteristics of mass-storage devices 12.8 Discuss operating-system services provided for mass storage, including RAID and HSM 12.9 Provide a high-level overview of distributed systems and the networks that interconnect them 12.10 Discuss the general structure of distributed operating systems 12.11 Explain the naming mechanism that provides location transparency and independence 12.12 Describe the various methods for accessing distributed files 12.13 Contrast stateful and stateless distributed file servers 12.14 Show how replication of files on different machines in a distributed file system is a useful redundancy for improving availability 12.15 Introduce the Andrew file system (AFS) as an example of a distributed file system
13 Describe an overview of operating systems. Define fundamental parts of an operating system.	13.1 Describe the services of an operating system 13.2 Describe the relationship between hardware and the operating system 13.3 Describe single job processing 13.4 Multitasking vs multiprocessing 13.5 Define concurrent processing vs simultaneous processing 13.6 Describe file management features 13.7 Describe scheduling 13.8 Describe the different types of operating systems 13.9 Describe the services an operating system provides to users, processes, and other systems 13.10 Discuss the various ways of structuring an operating system

14 Define the process of loading and executing a program	13.11 Explain how operating systems are installed and customised and how they boot 14.1 Develop a description of deadlocks, which prevent sets of concurrent processes from completing their tasks 14.2 Present a number of different methods for preventing or avoiding deadlocks in a computer system. 14.3 Describe basic scheduling concepts 14.4 Describe CPU scheduling decisions 14.5 Explore the several steps users' go through before being run 14.6 Define process swapping 14.7 Identify the characteristics of multimedia data 14.8 Examine several algorithms used to compress multimedia data 14.9 Explore the operating system requirements of multimedia data, including CPU and disk scheduling and network management 14.10 Define how virtual memory is implemented 14.11 Define demand paging
15 Describe the file system structure. Define file operations.	15.1 Explain the function of file systems 15.2 Describe the interfaces to file systems 15.3 Discuss file-system design tradeoffs, including access methods, file sharing, file locking, and directory structures 15.4 Explore file-system protection 15.4 Describe the details of implementing local file systems and directory structures 15.5 Describe the implementation of remote file systems 15.7 Discuss block allocation and free-block algorithms and trade-offs
16 Define how to protect the system resources. Describe the external environment of a system.	16.1 Discuss security threats and attacks 16.2 Explain the fundamentals of encryption, authentication, and hashing 16.3 Examine the uses of cryptography in computing 16.4 Describe the various countermeasures to security attacks

Recommended Learning Resources: Computer Systems Architecture

	• Introduction to Computing Systems: From Bits and Gates to C and Beyond 2nd Edition. ISBN 10: 0072467509

Text Books	• Computer Organization and Design Fundamentals by David Tarnoff ISBN: 978-1-4116-3690-3 • Principles of Computer Architecture Miles Murdocca and Vincent Heuring ISBN-10: 0201436647 Operating System Concepts, 8th Edition Abraham Silberschatz. ISBN 978-0-470-12872-5
Study Manuals 	BCE produced study packs
CD ROM 	Power-point slides
Software 	None