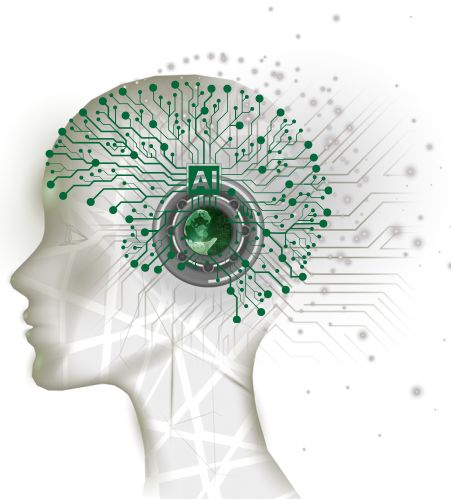




ZIAUDDIN UNIVERSITY
EXAMINATION BOARD

SSC A Computer Science Syllabus



For exams in 2026 & onwards

INTRODUCTION TO ZUEB

The Ziauddin University Examination Board (ZUEB) is not only an awarding body but also a solution-driven educational organization dedicated to upholding the highest standards of academic excellence. ZUEB believes in Excellence, Integrity, and Innovation in Education. Established with a vision to foster a robust educational environment, ZUEB is committed to nurturing intellectual growth and development that meets international standards in an effective manner. The Ziauddin University Examination Board (ZUEB) was established through Government Gazette No. XLI on June 6th, 2018. Its purpose is to ensure a high quality, maintain global standards, and align the syllabi with national integrity within the examination system of Pakistan. ZUEB manages student appeals, regulates assessments, and reviews policies to maintain high standards.

WHY CHOOSE SSC-A AT ZUEB?

Ziauddin University Examination Board (ZUEB) offers the SSC-A (Secondary School Certificate advance) program, designed for students from international educational backgrounds. This program provides a structured, affordable, and academically strong pathway for learners to align with Pakistan's education system. It allows students to fulfil national curriculum requirements, including Urdu, Islamiyat, Pakistan Studies, or Sindhi, with academic integrity and flexible learning options. ZUEB believes no student should be left behind due to financial limitations or cross-system transitions, and SSC-A serves as a bridge between past efforts and future ambitions. It is the trusted choice for higher education in Pakistan.

SSC-ADVANCE COMPUTER SCIENCE

Computer Science in the SSC-Advance qualification at ZUEB is a cornerstone subject for students aspiring to pursue careers in software engineering, information technology, data science, and related fields. It provides the essential foundation for computational thinking, problem-solving, and logical analysis — skills that are critical for academic excellence and intellectual growth. This subject not only strengthens understanding of digital systems but also equips students with the prerequisites required for success in competitive university entrance examinations across Pakistan.

Aligned with both national educational frameworks and the needs of students from international qualification backgrounds, our SSC-Advance Computer Science creates meaningful links between global digital knowledge and local academic standards. Students gain a comprehensive grasp of fundamental concepts in programming, algorithms, data structures, computer hardware, networking, and databases, delivered through a structured, flexible, and supportive learning model.

Whether your goal is to enter a top university for computer science, pursue software development, artificial intelligence, or cybersecurity, or simply build a strong foundation in computational and analytical reasoning, SSC-Advance Computer Science ensures you are academically prepared and nationally aligned. Explore more on what SSC-A offers: [ZUEB SSC-A Official Page](#).

Syllabus Overview			
No.	Content	AO	Exam
1	Systems Architecture	1,2,3	Combination of written exam papers (externally set and marked) Paper 1: Short Answer and Data Response Duration: 2 hours 20 minutes Paper 2: Problem Solving and Programming Duration: 1 hour 40 minutes
2	Algorithms, Programming and Logic	1,2,3	
3	Data representation	1,2,3	

Description of Assessment Objectives

A01 – Knowledge and Understanding

Exhibit comprehension and awareness of the foundational theories and principles of computer science, such as abstraction, logical reasoning, algorithms, and methods of representing data.

A02 – Application of Knowledge and Understanding

Demonstrate the ability to apply understanding of computer science concepts by analysing computational problems, identifying suitable strategies, and employing appropriate techniques to propose effective solutions.

A03 – Problem Solving and Evaluation

Design, implement, and critically assess computer-based systems, refining them through testing, evaluating their effectiveness, and making well-justified decisions leading to informed conclusions.

Weighting of Assessment Objectives

Assessment Objective	P1 (%)	P2 (%)
A01	50	30
A02	20	20
A03	30	50

Systems Architecture				
Aim: To enable learners to demonstrate a theoretical and practical understanding of computing systems.				
	The learner will:	SLO #	Assessment Criteria - The learner can:	Cognitive levels
1	Understand the significance of Systems architecture	1.1.1	Explain the purpose of the CPU.	AO2
		1.1.2	Examine Von Neumann architecture/ MAR (Memory Address Register)/ MDR (Memory Data Register)/ Program Counter/ Accumulator.	AO2
		1.1.3	Explore ALU (Arithmetic Logic Unit), CU (Control Unit), and Cache.	AO2
		1.1.4	Investigate the CPU's role in fetching and executing instructions stored in memory.	AO2
		1.1.5	Assess how CPUs affect their performance, considering factors such as clock speed, cache size, and the number of cores.	AO2
		1.1.6	Discuss the purpose of embedded systems.	AO1
		1.1.7	Describe microprocessors and their functions.	AO1
2	Understand the theoretical demonstration of Memory	1.2.1	Describe the purpose of ROM in a computer system.	AO1
		1.2.2	Describe the purpose of RAM in a computer system.	AO1
		1.2.3	State the difference between RAM and ROM.	AO1
		1.2.4	Describe virtual memory, cloud, secondary, and primary storages.	AO1
		1.2.5	Describe flash memory.	AO1
3	Understand the principles of data Storage	1.3.1	Compare common types of storage: optical, magnetic, and solid state.	AO2
		1.3.2	Evaluate the advantages and disadvantages of optical, magnetic, and solid-state storage in terms of capacity, speed, portability, durability, reliability, and cost.	AO2
4	Understand the principles of Wired and Wireless networks	1.4.1	Examine LAN (Local Area Network) and WAN (Wide Area Network).	AO2

		1.4.2	Compare client-server and peer-to-peer networks.	AO2
		1.4.3	Explore DNS (Domain Name Server) and cloud hosting.	AO2
		1.4.4	Analyse virtual networks.	AO2
5	Understand the concepts of Network topologies, protocols and layers	1.5.1	Explain star and mesh network topologies.	AO1
		1.5.2	Outline common network protocols including: • TCP/IP (Transmission Control Protocol/Internet Protocol) • HTTP (Hyper Text Transfer Protocol) • HTTPS (Hyper Text Transfer Protocol Secure) • FTP (File Transfer Protocol) • POP (Post Office Protocol) • IMAP (Internet Message Access Protocol) • SMTP (Simple Mail Transfer Protocol)	AO1
		1.5.3	Explain packet switching.	AO1
		1.5.4	Explain encryption.	AO1
		1.5.5	Compare symmetric and asymmetric encryption.	AO2
		1.5.6	Distinguish between a network interface card (NIC) and a media access control (MAC) address.	AO1
		1.5.7	Examine the purpose of an Internet Protocol (IP) address and classify the types of IP.	AO2
		1.5.8	Outline the functions of a router in a network.	AO1
		1.5.9	Discuss the purpose and functions of a web browser, including: • storing bookmarks and favourites • recording user history • allowing use of multiple tabs • storing cookies • providing navigation tools • providing an address bar	AO3
6	Understand the principles of System and cyber security	1.6.1	Identify and explain various forms of cyberattacks, including: • phishing, • brute force attacks, • poor network policy, • data interception, • distributed denial of service (DDoS) attack, • hacking, • malware (virus, worm, Trojan horse, spyware, adware, ransomware), • phishing, and • social engineering	AO1

		1.6.2	Explain network security measures to prevent vulnerabilities, including: • network policies, • firewalls, • user access levels, • anti-malware (anti-virus and anti-spyware), • authentication methods (username and password, biometrics, two-step verification), • automating software updates, • checking the spelling and tone of communications, • hecking the URL attached to a link, • privacy settings, • proxy-servers, and • the Secure Socket Layer (SSL) security protocol	AO1
7	Understand the principles of Ethical, legal, and environmental concerns	1.7.1	Compare and explain open source and proprietary software, highlighting their characteristics, advantages, and disadvantages.	AO1
		1.7.2	Explain the environmental impacts of computers, including e-waste, energy consumption, and resource usage.	AO1
8	Understand the concept and process of Digital Currencies and Blockchain transactions	1.8.1	Research and explain different types of electronic currencies and their practical applications.	AO2
		1.8.2	Research and explain how blockchain functions as a digital ledger and how it is used to track digital currency transactions.	AO2
9	Understand automated and emerging technologies	1.9.1	Describe how sensors, microprocessors and actuators can be used in collaboration to create automated systems.	AO2
		1.9.2	Explain the advantages and disadvantages of an automated system used for a given scenario such as industry, transport, agriculture, weather, gaming, lighting, and science.	AO2
		1.9.3	Explain the characteristics of a robot, factory equipment, domestic robots and drones, a mechanical structure or framework, electrical components, such as sensors, microprocessors and actuators, and programmable robots.	AO1
		1.9.4	Describe the roles that robots can perform and explain the advantages and disadvantages of their use.	AO2
		1.9.5	Explain the main characteristics of Artificial Intelligence (AI), basic operation and components of AI systems to simulate intelligent behaviour as it relates to expert systems and machine learning.	AO1
10	Understand computer input and output devices	1.10.1	Describe an input device and why it is required, including: • barcode scanner • digital camera • keyboard • microphone • optical mouse • QR code scanner • touch screen (resistive, capacitive and infra-red) • two-dimensional (2D) and three-dimensional (3D) scanners	AO1
		1.10.2	Describe an output device and why it is required, including: • actuator • digital light processing (DLP) projector • inkjet printer • laser printer • light emitting diode (LED) screen • liquid crystal display (LCD) projector • liquid crystal display (LCD) screen • speaker • 3D printer	AO1

		1.10.3	Describe sensors, their purposes, and types of data captured by each sensor, including: • acoustic • accelerometer • flow • gas • humidity • infra-red • level • light • magnetic field • moisture • pH • pressure • proximity • temperature	AO1
11	Understand the types of software and interrupts	1.11.1	Differentiate between system software and application software and provide examples of each.	AO1
		1.11.2	Outline the role and basic functions of an operating system (OS), including: • managing files • handling interrupts • providing an interface • managing peripherals and drivers • managing memory • managing multitasking • providing a platform for running applications • providing system security • managing user accounts	AO1
		1.11.3	Outline the role and operation of interrupts.	AO1
		1.11.4	Compare high-level language and a low-level language, including the advantages and disadvantages of each.	AO1
		1.11.5	Summarise the role of an integrated development environment (IDE) in writing program code and the common functions IDEs provide, including: • code editors • run-time environment • translators • error diagnostics • auto-completion • auto-correction • prettyprint	AO1

Algorithms, Programming and Logic				
<i>Aim: To enable students to demonstrate a theoretical and practical understanding of computational thinking, algorithms, and programming.</i>				
	The learner will:	SLO #	Assessment Criteria - The learner can:	Cognitive levels
1	Understand the principles of computational thinking	2.1.1	Analyse computational thinking and its key elements, including: • abstraction • decomposition	AO3
		2.1.2	Explain the use of searching algorithms, including: • binary search • linear search	AO2
		2.1.3	Explain the use of sorting algorithms, including: • bubble sort • merge sort • insertion sort	AO2
		2.1.4	Evaluate the use of, pseudo code vs flow diagrams.	AO3
		2.1.5	Analyse the standard methods of solution such as: • totalling • counting • finding maximum • minimum and average values	AO3
2	Understand Programming techniques	2.2.1	Analyse programming constructs used to control the flow of a program: • sequence • selection • iteration (count and condition controlled loops)	AO3
		2.2.2	Analyse programming constructs used to control basic file handling operations: • open • read • write • close	AO3
		2.2.3	Analyse programming constructs used to control basic one- and two-dimensional arrays.	AO3
		2.2.4	Analyse programming constructs used to control arithmetic operators and the common Boolean operators.	AO3
		2.2.5	Analyse programming constructs used to control basic data types: • integer • real • Boolean • character and string • casting	AO3
		2.2.6	Analyse programming constructs used to control how to identify syntax and logic errors.	AO3

		2.2.7	Analyse the purpose of testing.	AO3
3	Understand how to create a maintainable programs and arrays	2.3.1	Develop a maintainable program using appropriate and meaningful identifiers such as the commenting feature provided by the programming language, procedures and functions, relevant and appropriate commenting of syntax, variables, constants, and arrays.	AO3
		2.3.2	Implement and use one-dimensional (1D) and two-dimensional (2D) arrays.	AO3
		2.3.3	Manipulate values in an array by writing into and reading from it using iteration.	AO3
4	Understand the principles of computational logic	2.4.1	Analyse logic diagrams using the operations AND, OR, NOT, NAND, NOR, XOR (EOR), determine the binary output for all possible binary inputs, with all other gates limited to two inputs.	AO2
		2.4.2	Analyse combinations of Boolean operators using AND, OR, NOT, NAND, NOR, XOR (EOR), determine the binary output for all possible binary inputs, with all other gates limited to two inputs.	AO2
		2.4.3	Evaluate the use of truth tables.	AO3
		2.4.4	Construct a Boolean logic expression from a: • problem statement • logic circuit • truth table	AO3
5	Understand the program development life cycle	2.5.1	Describe the analysis, design, coding, and testing stages.	AO
		2.5.2	Implement structure diagrams, abstraction, decomposition, pseudocodes, flowcharts, problem and requirement identifications, and design, coding, and testing techniques for test data.	AO
6	Understand databases and data types	2.6.1	Describe a single-table database from given data storage requirements, including: • fields • records • validation	AO1
		2.6.2	Analyse suitable basic data types, including: • text/alphanumeric • character • Boolean • integer • real • date/time	AO3
		2.6.3	Analyse the purpose of a primary key and determine a suitable primary key for a given database table.	AO3

		2.6.4	Analyse and complete structured query language (SQL) scripts to query data stored in a single database table, including: • SELECT • FROM • WHERE • ORDER BY DESCENDING • ORDER BY ASCENDING • SUM • COUNT • AND • OR	AO3
--	--	-------	---	-----

Data representation				
Aim: To enable students to understand data representation.				
	The learner will:	SLO #	Assessment Criteria - The learner can:	Cognitive levels
1	Understand the purpose of Translators and facilities of languages	3.1.1	Evaluate the purpose of translators.	AO3
		3.1.2	Describe the assembler, a compiler and an interpreter.	AO1
2	Understand data representation	3.2.1	Explain types of compression, including: • lossy • lossless	AO2
		3.2.2	Explain the effect of colour depth and resolution on the size of an image file.	AO2
		3.2.3	Explain how an image is represented as a series of pixels in binary values.	AO2
		3.2.4	Explain the effect of colour depth and resolution on the size of an image file.	AO2
		3.2.5	Identify and define bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte, and exbibyte (EiB).	AO1
3	Understand the denary, binary and hexadecimal number systems	3.3.1	Convert between: • positive denary and positive binary • positive denary and positive hexadecimal • positive hexadecimal and positive binary	AO2
		3.3.2	Calculate the sum of two positive 8-bit binary integers.	AO2
		3.3.3	Calculate and identify overflow in binary addition.	AO2
		3.3.4	Apply a logical binary shift to a positive 8-bit binary integer.	AO2
		3.3.5	Apply the two's complement system to represent positive and negative 8-bit binary integers.	AO2
		3.3.6	Explain why a computer represents text and describe the use of character sets, including American standard code for information interchange (ASCII) and Unicode.	AO2
		3.3.7	Explain why a computer represents sound and describe the effects of sample rate and sample resolution.	AO2
		3.3.8	Explain why a computer represents an image and describe the effects of resolution and colour depth.	AO2

		3.3.9	Calculate the file size of an image and a sound file using provided data.	AO2
4	Understand types and methods of data transmission	3.4.1	Explain the structure of a packet.	AO1
		3.4.2	Explain the process of packet switching.	AO2
		3.4.3	Explain how data is transmitted from one device to another using different methods of data transmission, including: • serial • parallel • simplex • half-duplex • full-duplex	AO2
		3.4.4	Assess the suitability of each method of data transmission.	AO2
		3.4.5	Describe how the USB interface is used for data transmission.	AO2
5	Understand methods of error detection	3.5.1	Explain the processes involved in each of the following methods used for detecting errors in data after transmission: • parity check (odd and even) • checksum • echo check • parity byte and parity block check	AO1
		3.5.2	Explain how a check digit is used to detect errors in data entry and identify examples of its use, including international standard book numbers (ISBN) and bar codes.	AO2
		3.5.3	Explain how an automatic repeat query (ARQ) can be used to establish that data is received without error, including: • positive/negative acknowledgements • timeout	AO2