



College of Computing  
Department of Computer Science

*Innovative Learning Competencies Indicators  
Guiding Document*

For Bachelor of Science Degree (B.Sc.)  
In Computer Science

June 2022

Debre Berhan  
Ethiopia



## Contents

Introduction.....	1
Learning domains (Cognitive, Psychomotor & Affective).....	1
Professional Profile.....	2
Graduate Profile .....	2
List of Modules for Compulsory Courses.....	4
The core competency of compulsory courses.....	5
Sample Examination .....	36
Answer Key .....	47



## Introduction

The quality of higher education is considered one of the major elements that guarantee sustainable economic and social development at the national and regional levels of a given economy. There are different mechanisms that can be used to improve the quality of higher education. One of the mechanisms of ensuring the quality of higher education is to test the competence of candidates (prospective graduates) before graduation through the university exit examination. University Exit Exam is a standardized comprehensive curriculum-based that is designed to assess if students have achieved the minimum competence that was stipulated in the graduates' profile.

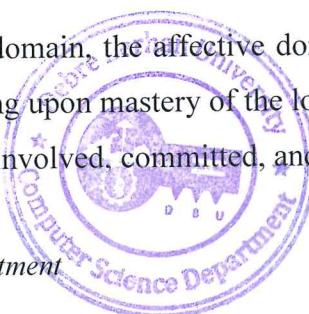
The purpose of this document is to present the **major courses** and their respective **learning content** and **competencies** of the computer science curriculum for prospective graduates. Hence, prospective computer science graduates are expected to be equipped with **core learning competencies** and prepare themselves for the exit examination.

## Learning domains (Cognitive, Psychomotor & Affective)

**Cognitive Domain:** According to researchers, there are six levels of cognitive complexity: knowledge, comprehension, application, analysis, synthesis, and evaluation. In the chart, as shown below [[Cognitive, Affective, and Psychomotor Domains \(washington.edu\)](#)], note the hierarchical arrangement, which means that higher levels subsume ability in lower levels? The higher the level, the presumably more complex mental operation is required. Higher levels are not necessarily more desirable than lower levels, because one cannot achieve the higher levels without an ability to use the lower levels. As one moves up into higher levels, however, the more applicable the skills are to those needed in daily life.

**Psychomotor Domain:** This domain is given primarily for information. Other courses within the curriculum stress the various levels of psychomotor performance (e.g., Laboratory Skills, programming practice). Psychomotor behaviors are performed actions that are neuromuscular in nature and demand certain levels of physical dexterity.

**Affective Domain:** Like the cognitive domain, the affective domain is hierarchical with higher levels being more complex and depending upon mastery of the lower levels. With the movement to more complexity, one becomes more involved, committed, and self-reliant.



## **Professional Profile**

Computer science professionals must have

- ✓ A high-level understanding of computing technologies as a system of wholeness. This understanding must go beyond the implementation details of the various components to encompass an appreciation for the structure of computer systems and the processes involved in their construction and analysis.
- ✓ Thorough understanding of the balance between theory and practice and the essential link between them, means it is not only the subject matter but also how the theoretical stances of the discipline influence.
- ✓ A solid foundation that allows and encourages prospective graduates to maintain their skills as the field evolves.

## **Graduate Profile**

The graduate will have the following knowledge, skill, and attitude at the end of the successful completion of the courses of the program.

### **A) Cognitive knowledge and skill**

- ✓ Demonstrate knowledge and understanding of essential facts, concepts, principles, and theories relating to computer science.
- ✓ Use such knowledge and understanding in the modeling and designing of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
- ✓ Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.
- ✓ Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.
- ✓ Deploy appropriate theories, practices, and tools for the specification, design, implementation, and evaluation of computer-based systems.
- ✓ Recognize the need for and an ability to engage in continuing professional development and the knowledge and skills gap fulfillment strategy to act as research assistants or lecturers in higher education institutions.

## **B) Practical skills**

- ✓ Specify, design, and implement computer-based systems.
- ✓ Evaluate systems in terms of general quality attributes and possible trade-offs presented within the given problem.
- ✓ Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, and multimedia systems.
- ✓ Identify any risks or safety aspects that may be involved in the operation of computing equipment within a given context.
- ✓ Deploy effectively the tools used for the development and documentation of application, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.
- ✓ Use current techniques, skills, and tools necessary to maintain and administer computer-based systems.
- ✓ Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and designing of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

## **C) Attitude (Transferable skill)**

- ✓ Make succinct presentations to a range of audiences about technical problems and their solutions.
- ✓ Be able to work effectively as a member of a development team.
- ✓ Understand and explain the both qualitative and quantitative dimensions of a problem.
- ✓ Manage one's own learning and development, including the overall system development and implementation of project management skills.
- ✓ Keep in touch with the current state-of-the-art system development strategy in the discipline to continue one's own professional development.
- ✓ Recognize and be guided by the social, professional, and ethical principles involved in the development and use of computer technology.



## List of Modules for Compulsory Courses

Course Category	Code	Courses Under Category
<b>Basic programming and Emerging Technologies</b>	01	Computer programming Fundamentals of Programming in C++
<b>Computer Architecture and Operating Systems</b>	02	Computer organization and architecture
		Operating System
		Microprocessor and Assembly Language Programming
		Real time and embedded system
<b>Computer Networking and Security</b>	03	Data Communication & Computer Networking
		Wireless Communication and Mobile Computing
		Computer Security
		Network and System Administration
		Introduction to Distributed Systems
<b>Database Systems</b>	04	Fundamentals of Database Systems
		Advanced Database Systems
<b>Advanced Programming</b>	05	Object-Oriented Programming
		Java Programming
<b>System Development</b>	06	Software Engineering
<b>Computer Graphics and HCI</b>	07	Computer Graphics
<b>Web and Application Development</b>	08	Web programming
		Design and Analysis of Algorithms
<b>Algorithms</b>	09	Data Structures and Algorithms
		Automata and Complexity Theory
<b>Compiler Development and Complexity</b>	10	Compiler Design
		Introduction to Artificial Intelligence
<b>Intelligent Systems</b>	11	Computer Vision and Image Processing
		Research Methods in Computer Science
<b>Projects and Research</b>	12	Selected topics in Computer Science
<b>Selected topics</b>	13	Selected topics in Computer Science

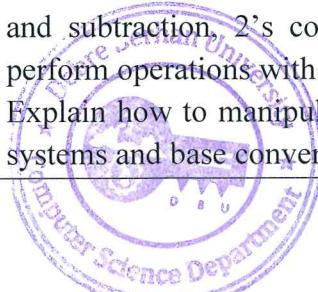


## The core competency of compulsory courses

Upon the successful completion of the course, students should be able to perform all the learning competencies of the courses specified here:

N o	Course Name	Learning Content	Learning competencies based on the three Learning domains (Cognitive, Psychomotor & Affective)
1	Computer Programming	<b>Problem-Solving &amp; Algorithm Development</b> <ul style="list-style-type: none"> <li>• Flowchart</li> <li>• Pseudo code</li> <li>• Algorithm</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain system development modeling tools</li> <li>✓ Analyze a variety of problems and generate appropriate algorithmic solutions</li> <li>✓ Discuss the different types of paradigms in programming</li> <li>✓ Formulate programs and design flowcharts to solve problems, and explain with pseudo-codes and algorithms</li> </ul>
		<b>Basics of Programming</b>	<ul style="list-style-type: none"> <li>✓ Construct the basic element of programming language (Keywords, identifiers literals, comments, variables data types, constant)</li> <li>✓ Formulate computer programs that help to solve real-life problems using programming concepts</li> </ul>
		<b>Control Statements</b>	<ul style="list-style-type: none"> <li>✓ Define the different control statements with their respective applicability</li> <li>✓ Try to solve problems using the concepts of control statements, (If...Else; If... else if... else; Switch; Loop (i.e. For, While, Do...while); Jumping Statements.</li> </ul>
		<b>Array, String, and Function</b>	<ul style="list-style-type: none"> <li>✓ Discuss about the declaration, initialization, and the use of different dimensions of an array</li> <li>✓ Describe different string manipulation operations to solve the problem</li> <li>✓ Formulate functions with function overloading, default parameter, scope operator, pass arguments by value and reference in a program</li> <li>✓ Apply your knowledge to solve problems using an array, string, function and declaration of variables</li> </ul>
2	Fundamentals of Programming in C++	<b>Functions in C++</b>	<ul style="list-style-type: none"> <li>✓ Distinguish between parameter and argument and apply accordingly</li> <li>✓ Identify elements of OOP Elements</li> </ul>
		<b>Arrays</b>	<ul style="list-style-type: none"> <li>✓ Manipulate and access a multi-dimensional array</li> </ul>

	<b>String and Pointer</b>	<ul style="list-style-type: none"> <li>✓ Work with functions, strings, arrays, and pointers to solve real-world problems wherever they are applicable</li> <li>✓ Explain and use reference and dereference operators with pointers and manipulate pointer arithmetic</li> <li>✓ Create programs that use array and String data structures</li> </ul>	
	<b>Structure in C++</b>	<ul style="list-style-type: none"> <li>✓ Describe simple and nested structures in C++ program</li> <li>✓ Discuss Nested Structure Program in C++</li> </ul>	
	<b>File &amp; File Management</b>	<ul style="list-style-type: none"> <li>✓ Explain and utilize function overloading and recursion in programming</li> <li>✓ Describe and implement file management concepts in programming, operations like open(), read(), write() and close().</li> <li>✓ Write a complete program using C++ that includes all file classes and operations.</li> </ul>	
3	<b>Computer organization and Architecture</b>	<p><b>Introduction to Computer Organization &amp; Architecture</b></p> <ul style="list-style-type: none"> <li>• Logic gates</li> <li>• Boolean Algebra</li> <li>• Combinational circuits</li> <li>• Flip-Flops</li> <li>• Sequential circuits</li> </ul>	<ul style="list-style-type: none"> <li>✓ Discuss the fundamental of computer organization &amp; architecture</li> <li>✓ Explain Boolean and logical operators, basic digital logic circuits</li> <li>✓ Demonstrate how to design simple circuits using digital logic gates</li> <li>✓ Describe the basic building blocks and their roles in the development of computer architecture</li> <li>✓ Explain the different types of flip flops</li> <li>✓ Compare and contrast combinational and sequential circuits with examples</li> <li>✓ Explain how to apply mathematical expressions to describe the functions of simple combinational and sequential circuits</li> </ul>
	<b>Number System</b>	<ul style="list-style-type: none"> <li>• Data types</li> <li>• Complements</li> <li>• Fixed- and floating-point representation</li> <li>• Codes</li> </ul>	<ul style="list-style-type: none"> <li>✓ Discuss the different data type representation</li> <li>✓ Explain the operation of the arithmetic unit including the algorithms &amp; implementation of fixed-point and floating-point addition, subtraction, multiplication &amp; division</li> <li>✓ Discuss the different number systems, binary addition and subtraction, 2's complement representation and perform operations with these representations</li> <li>✓ Explain how to manipulate between different number systems and base conversion</li> </ul>

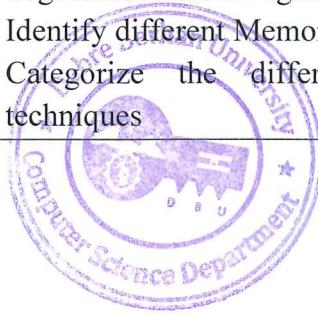


	<p><b>Common digital components</b></p> <ul style="list-style-type: none"> <li>• Integrated circuit</li> <li>• Decoder, multiplexer and registers</li> <li>• Binary counter</li> <li>• Memory units</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain Integrated circuit</li> <li>✓ Describe the common digital components such as decoder, multiplexer, register, memory, etc.)</li> <li>✓ Explain how to simplify logic expressions or Boolean functions using K-map simplification</li> <li>✓ Discuss binary counter and memory units</li> </ul>
	<p><b>Register Transfer Language and Micro Operations</b></p> <ul style="list-style-type: none"> <li>• RTL</li> <li>• Bus and memory transfer</li> <li>• Micro operation <ul style="list-style-type: none"> <li>◦ Arithmetic, logic, Shift micro-operations</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain Register Transfer Language</li> <li>✓ Explain the memory read and write operations (using transfer statements)</li> <li>✓ Define the bus and describe how the bus is constructed with multiplexers</li> <li>✓ Mention and discuss the different types of micro-operations performed on data stored in registers</li> <li>✓ Discuss how to perform arithmetic, logic, and shift micro-operations on data stored in registers.</li> </ul>
	<p><b>Basic Computer Organization and Design</b></p> <ul style="list-style-type: none"> <li>• Instructional code</li> <li>• Computer Register</li> <li>• Computer Instructions</li> <li>• Timing and control</li> <li>• Memory reference instructions</li> <li>• Design of Basic computers</li> <li>• Design of accumulator logic</li> </ul>	<ul style="list-style-type: none"> <li>✓ Discuss the different performance improvement measures of computer architecture</li> <li>✓ Explain the basic structure of computer hardware &amp; software</li> <li>✓ Discuss basic concepts of circuits and their design aspects</li> </ul>
	<p><b>Central Processing Unit</b></p> <ul style="list-style-type: none"> <li>• General register organization</li> <li>• Stack organization</li> <li>• Instruction formats</li> <li>• Addressing modes</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain the architecture and functionality of the CPU</li> <li>✓ Identifying the major components of the CPU (e.g. instruction/data cache, registers, ALU).</li> <li>✓ Identify the processes involved in the basic operations of the CPU</li> <li>✓ Discuss principles of instruction set design, including RISC architectures</li> </ul>

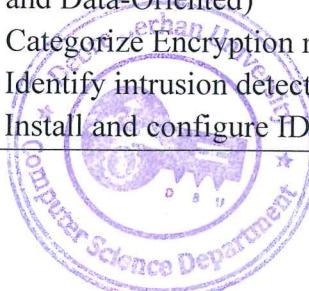


		<ul style="list-style-type: none"> <li>• Data transfer and manipulation</li> <li>• Program control</li> <li>• Characteristics of RISC and CISC</li> </ul>	<ul style="list-style-type: none"> <li>✓ Identify the elements of modern instruction sets and their impacts on processor design</li> <li>✓ Describe the instruction set of a modern RISC and CISC processors and their characteristics</li> </ul>
		<p><b>Memory and Input-Output Organization</b></p> <ul style="list-style-type: none"> <li>• Memory Hierarchy</li> <li>• Main memory</li> <li>• Cache memory</li> <li>• Mapping functions</li> <li>• External memory <ul style="list-style-type: none"> <li>◦ Magnetic disks</li> <li>◦ RAID technology</li> <li>◦ Optical disks</li> <li>◦ Magnetic tapes</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>✓ Describe memory system organization and architecture</li> <li>✓ Explain the principles of memory management</li> <li>✓ Mention the main types of memory technology</li> <li>✓ Explain the function of each element of a memory hierarchy</li> <li>✓ Identify and compare different methods for the computer I/O</li> <li>✓ Discuss how memory is organized and managed in a modern digital computer, including the virtual and physical memory</li> <li>✓ Compare the different types of memory in terms of speed and storage capacity</li> <li>✓ Describe the role of cache and virtual memory</li> <li>✓ Explain types and technologies of external memory</li> <li>✓ Discuss input-output units, including in particular hard disks and solid-state disks (SSDs), how they communicate with the processor, and how their performance is calculated</li> </ul>
		<p><b>Input and output devices</b></p> <ul style="list-style-type: none"> <li>• Peripheral Devices</li> <li>• I/O interface</li> <li>• Mode of transfer</li> <li>• DMA</li> <li>• IOC</li> <li>• Serial communication</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain the terms I/O interfaces and I/O processor.</li> <li>✓ Describe why input and output devices cannot be connected directly with the CPU.</li> <li>✓ Discuss the different ways (modes of data transfer) by which data can be sent from the I/O device to the memory by the CPU?</li> <li>✓ Explain how interrupts are used to implement I/O control and data transfers.</li> <li>✓ Explain the DMA and IOC</li> <li>✓ Describe Serial communication</li> </ul>
		<p><b>Pipeline and Vector Processing</b></p>	<ul style="list-style-type: none"> <li>✓ Describe pipelining and parallelism features applied in a single processor, multiprocessors and multicore architectures</li> <li>✓ Discuss how pipelining in a processor function</li> <li>✓ Explain multiprocessor and its characteristics</li> </ul>
4	<b>Operating Systems</b>	<b>Introduction to Operating Systems</b>	<ul style="list-style-type: none"> <li>✓ Define the Operating system and discuss its roles and purposes</li> </ul>

	<ul style="list-style-type: none"> <li>• Role and purpose of operating systems</li> <li>• History of operating system development</li> <li>• Types of operating systems</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain the following Operating system concepts: Context Switching, Multiprogramming, System Calls, Process, Threads, Files, Deadlocks, Device Drivers</li> <li>✓ Describe the relationship between Operating Systems and Computer Hardware</li> <li>✓ Characterize and identify the types of Operating Systems for different purposes of computing</li> <li>✓ Able to Install an Operating System and understand the boot Process</li> </ul>
	<p><b>Processes and Process Management</b></p> <ul style="list-style-type: none"> <li>• Process and Thread</li> <li>• The concept of multi-threading</li> <li>• Inter-process communication</li> <li>• Process Scheduling</li> <li>• Deadlock</li> </ul>	<ul style="list-style-type: none"> <li>✓ Describe Process, Threads, Deadlocks, Multiprogramming, System Calls, Race Condition, Process Scheduling, Scheduling Algorithms, Mutual Exclusion Conditions, Deadlocks, Multithreading</li> <li>✓ Explain Process and Thread Management</li> <li>✓ Characterize Types of Solutions for Race Conditions, Busy Waiting, Sleep and Wake Up using semaphores Message Passing</li> <li>✓ Identify preemptible able and non-pre-emptible devices</li> <li>✓ Identify which Scheduling algorithms are best used for different scenarios</li> <li>✓ Scheduling processes based on arrival time and CPU cycles</li> <li>✓ Able to Model Deadlocks</li> <li>✓ Identify Deadlocks using deadlock detection algorithms</li> <li>✓ Perform dynamic avoidance using safe and unsafe states</li> <li>✓ Create, Manage, Kill processes and threads in JAVA</li> <li>✓ Develop process scheduling algorithms</li> </ul>
	<p><b>Memory Management</b></p> <ul style="list-style-type: none"> <li>• Physical memory and memory management</li> <li>• Paging and Segmentation</li> <li>• Caching</li> </ul>	<ul style="list-style-type: none"> <li>✓ Define memory management</li> <li>✓ Explain the basic tasks of a memory management system (Protection and Sharing)</li> <li>✓ Identify the creation of executable code</li> <li>✓ Describe Address Translation, Swapping, Cache</li> <li>✓ Discuss Fragmentation, Partition techniques such as Segmentation and Paging</li> <li>✓ Identify different Memory management schemes</li> <li>✓ Categorize the different types of partitioning techniques</li> </ul>



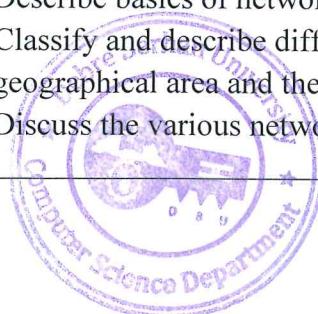
		<ul style="list-style-type: none"> <li>✓ Monitor memory usage for different process</li> <li>✓ Perform Efficient memory usage practices during programming</li> </ul>
	<b>Device Management</b> <ul style="list-style-type: none"> <li>• Characteristics of parallel and serial devices</li> <li>• Buffering strategies</li> <li>• Direct memory access</li> <li>• Recovery from failure</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain stream and Block Input/Output</li> <li>✓ Describe Buffering</li> <li>✓ Explain DMA</li> <li>✓ Identify different I/O management techniques</li> <li>✓ Categorizing the I/O Management (Memory Mapped I/O, Polled I/O)</li> <li>✓ Categorize Device classes (character, block, networking and miscellaneous)</li> <li>✓ Able to recover from failure and install and configure device drivers</li> </ul>
	<b>File Systems</b> <ul style="list-style-type: none"> <li>• Fundamental concepts on file</li> <li>• File system techniques</li> <li>• Special purpose file systems</li> </ul>	<ul style="list-style-type: none"> <li>✓ Describe Files, Data, and Meta Data</li> <li>✓ Explain Sequential and Random Data Access</li> <li>✓ Discuss Directories</li> <li>✓ Identify File systems (NTFS, FAT32, EXT....)</li> <li>✓ Discuss Partitioning</li> <li>✓ Explain file Naming</li> <li>✓ Identify File System Types</li> <li>✓ Categorize Data Types</li> <li>✓ Categorize File Systems (Journaling and Non-Journaling)</li> <li>✓ Identify which file system is best for different scenarios</li> <li>✓ Format file systems and install different file systems</li> <li>✓ Defragment and partition file systems</li> <li>✓ Mount and Un-mount File Systems</li> </ul>
	<b>Security and Protection</b> <ul style="list-style-type: none"> <li>• Policies and mechanism of system security</li> <li>• System protection, authentication</li> </ul>	<ul style="list-style-type: none"> <li>✓ Discuss Security, Protection, and Security Threats</li> <li>✓ Explain Protection Mechanisms (Access Control, Cryptography, Intrusion Detection)</li> <li>✓ Categorize security threats</li> <li>✓ Categorize computer system assets</li> <li>✓ Identify which computer system assets are affected by which security threats with examples</li> <li>✓ Categorize Protection strategies</li> <li>✓ Categorize Access control mechanisms (User-oriented and Data-Oriented)</li> <li>✓ Categorize Encryption mechanisms</li> <li>✓ Identify intrusion detection SW</li> <li>✓ Install and configure IDS</li> </ul>



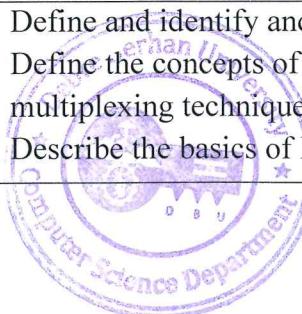
			<ul style="list-style-type: none"> <li>✓ Install and configure different types of access control mechanisms in windows and Linux</li> </ul>
5	<b>Microprocessor and Assembly Language Programming</b>	<b>Introduction to Microprocessor</b>	<ul style="list-style-type: none"> <li>✓ Discuss the evaluation of the microprocessor and its type</li> </ul>
		<b>8086 Microprocessor Architecture</b>	<ul style="list-style-type: none"> <li>✓ Discuss the architecture of the 8086 microprocessors</li> <li>✓ Identify registry, memory allocation, memory reference techniques, File processing and modular programming</li> <li>✓ Describe various registers of 8086 microprocessor and their functionality</li> <li>✓ Discuss the status of the 8086-microprocessor using the flag registers and how flag registers are affected by the instructions</li> <li>✓ Compare and contrast the minimum and maximum mode of 8086 hardware specification</li> <li>✓ Write an Assemble language program that indicates the effects of flag registers</li> </ul>
		<b>Addressing Modes and Instruction Set</b>	<ul style="list-style-type: none"> <li>✓ Classify and articulate the addressing mode of the 8086 microprocessor and memory access method</li> <li>✓ Illustrate and explain the kinds of addressing modes of the 8086 microprocessors</li> <li>✓ Discuss and manipulate data movement, arithmetic, and logical instructions</li> <li>✓ Write an ALP to show different types of addressing modes and instruction sets</li> </ul>
		<b>Program Control Instructions</b>	<ul style="list-style-type: none"> <li>✓ Explain the process of flow control in assembly language</li> <li>✓ Demonstrate assembly programming code to use the branching structures, looping or rotating structures, procedures, macros, and interrupts</li> <li>✓ Discuss how interrupts are used to implement I/O control and data transfers</li> <li>✓ Identify and apply the interrupt instruction, interrupt vectors, and their use</li> <li>✓ Mention the kinds of interrupted instructions</li> <li>✓ Explain how to write code to process exceptions and interrupts</li> </ul>
	<b>Memory and I/O Interfacing</b>		<ul style="list-style-type: none"> <li>✓ Describe how memory and I/O Interfacing work</li> <li>✓ Explain I/O Port Address Decoding</li> </ul>

			<ul style="list-style-type: none"> <li>✓ Identify how to interface serial and parallel I/O devices with a microprocessor</li> <li>✓ Demonstrate how Analog-to-Digital (ADC) and Digital-to-Analog (DAC) Interfacing work</li> </ul>
6	<b>Real-time and embedded system</b>	<b>Introduction to Real-Time and Embedded System</b>	<ul style="list-style-type: none"> <li>✓ Explain real-time systems and embedded systems</li> <li>✓ Discuss fundamentals of real-time Terminology</li> <li>✓ Explain Real-time design issues such as real-time machine organization and architecture</li> <li>✓ Classify the different real-time systems based on different use cases</li> <li>✓ Discuss examples of Real-time systems applications</li> </ul>
		<b>Embedded System Architecture</b>	<ul style="list-style-type: none"> <li>✓ Describe Hardware architectures for Embedded Systems</li> <li>✓ Mention Architecture Types (Von Neumann Architecture, Harvard Architecture, CISC, and RISC)</li> <li>✓ Discuss the Embedded Hardware organization using ARM Cortex Mo+</li> <li>✓ Explain the fundamentals of register I/O (Analog I/O for sensors)</li> <li>✓ Describe the Communication using parallel and serial ports in real-time systems</li> <li>✓ Discuss the primitives, register organization and programming for Assembly language Programming with ATmega32</li> <li>✓ Demonstrate how to a “C-language” program for interfacing with real-time systems</li> </ul>
		<b>Software Frameworks for Realtime and Embedded</b>	<ul style="list-style-type: none"> <li>✓ Discuss Real-time OS functionality, structure, Interfaces, and applications.</li> <li>✓ Explain the Real-time kernels concepts</li> <li>✓ Polled loop system, phase state-driven code</li> <li>✓ Explain terminologies such as co-routine interrupt, Inter-Process Communications</li> <li>✓ Explain Interrupt driven systems, foreground/background systems, full feature real-time operating system</li> <li>✓ Discuss how to perform real-time scheduling</li> <li>✓ Describe how to allocate resources to tasks and Inter-Task Communication</li> <li>✓ Discuss Synchronization: Buffering Data, Mailboxes, Critical</li> </ul>

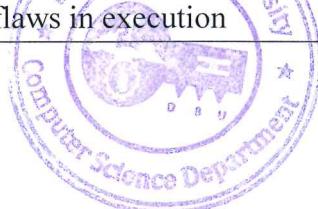
			<ul style="list-style-type: none"> <li>✓ Explain how Embedded OS issues Semaphores, Event flags and signals, Deadlock work</li> </ul>
		<b>Embedded Systems Design Issues</b>	<ul style="list-style-type: none"> <li>✓ Explain embedded system design concepts</li> <li>✓ Discuss the Memory management issues in embedded systems (Sharing of Memory and Memory) protection</li> <li>✓ Describe Hardware architectures and organization for embedded system</li> <li>✓ Discuss issues in Developing and real-time software</li> </ul>
		<b>Real-Time Communication</b>	<ul style="list-style-type: none"> <li>✓ Explain the fundamental constraints in real-time system communication and networking</li> <li>✓ Discuss communication options in real-time systems (LAN, ADHOC)</li> <li>✓ Discuss Quality of services issues in real-time systems</li> <li>✓ Explain the means to ensure the quality of services in real-time systems</li> <li>✓ Discuss Real-time systems applications in the Internet of things and wireless sensor networks</li> <li>✓ Explain and understand the impact of IPV6 in enabling the large-scale deployment of real-time systems</li> <li>✓ Evaluate, compare, and contrast different scheduling algorithms and real-time and embedded kernel designs</li> <li>✓ Apply knowledge of real-time and embedded system concepts to address an open research question in real-time and embedded systems and related fields</li> </ul>
7	<b>Data Communication and Computer Networks</b>	<b>Data Communication</b> <ul style="list-style-type: none"> <li>● <b>Computer Networks</b> <ul style="list-style-type: none"> <li>○ Network Applications</li> <li>○ Network Types</li> <li>○ Network Components</li> <li>○ Network Topology</li> <li>○ Network Models</li> </ul> </li> <li>● <b>Transmission Media</b> <ul style="list-style-type: none"> <li>○ Guided</li> <li>○ Unguided</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>✓ Define data communication and its fundamentals</li> <li>✓ Discuss the benefits and need for networks</li> <li>✓ List and describe different data communication components</li> <li>✓ Describe and characterize different modes of data transmissions</li> <li>✓ Classify the networking based on the different parameters</li> <li>✓ Mention and describe a protocol</li> <li>✓ Mention and describe protocol elements</li> <li>✓ Discuss the types of switching (i.e., circuit, packet)</li> <li>✓ Describe basics of networks</li> <li>✓ Classify and describe different networks based on geographical area and their architecture</li> <li>✓ Discuss the various network components</li> </ul>



		<ul style="list-style-type: none"> <li>✓ Explain the type of network topology (i.e. Physical and Logical)</li> <li>✓ Describe the OSI reference model and layer's function</li> <li>✓ Explain the type of transmission media</li> <li>✓ Discuss guided and unguided media and its type of communication</li> <li>✓ Discuss the function of the OSI layer</li> <li>✓ Identify and label services and protocols of each layer</li> <li>✓ Identify and describe the basics of the TCP/IP model</li> <li>✓ Mention and explain layers in TCP/IP model</li> <li>✓ Identify and describe the function of different network layers and protocols</li> <li>✓ Identify the address mechanisms used under the TCP/IP model</li> <li>✓ Describe basic concepts of IPv4 and IPv4 header format</li> <li>✓ Define addressing space and Classify IPv4 IP address</li> <li>✓ Calculate class full and classless IP address</li> <li>✓ Define and describe NAT and private IP address</li> <li>✓ Define routing, routing table, and routing protocols</li> <li>✓ Classify routing protocols and Name different routing algorithms</li> <li>✓ Compare and contrast static and adaptive routing</li> <li>✓ Identify and describe routing metrics</li> <li>✓ Contrast OSI and TCP/Ip models</li> <li>✓ Criticize both OSI and TCP/IP models</li> <li>✓ Identify and classify familiar network protocols</li> </ul>
	<b>Network types</b>	<ul style="list-style-type: none"> <li>✓ Classify networks based on different parameters</li> <li>✓ Describe and compare peer-to-peer and server-based networks</li> <li>✓ Compare and contrast packet switching and circuit switching networks</li> <li>✓ Identify and describe network cabling</li> <li>✓ Create and test network cables</li> <li>✓ Design different network topologies</li> <li>✓ Explain how to set up small-scale networks</li> </ul>
	<b>Switching and Multiplexing</b>	<ul style="list-style-type: none"> <li>✓ Define and identify and classify switching types</li> <li>✓ Define the concepts of multiplexing and classify multiplexing techniques</li> <li>✓ Describe the basics of Ethernet and its varieties</li> </ul>

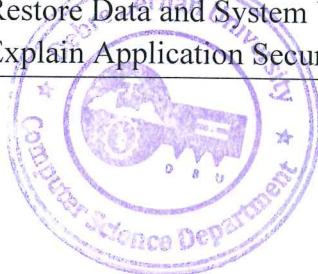


			<ul style="list-style-type: none"> <li>✓ Identify and describe basic concepts of wireless networks</li> </ul>
8	<b>Wireless Communication and Mobile Computing</b>	<b>Introduction to Wireless Communications</b>	<ul style="list-style-type: none"> <li>✓ Discuss different types of wireless communication services and applications</li> <li>✓ Explain the difference between wireless &amp; mobile</li> <li>✓ Describe the advantages of wireless communication as an alternative to fixed media</li> </ul>
		<b>Mobile Computing</b>	<ul style="list-style-type: none"> <li>✓ Describe Fundamentals of Mobile computing</li> <li>✓ Explain Mobile devices and Mobile OS</li> <li>✓ Discuss Mobile Computing Issues, Challenges, Technologies, and Applications</li> </ul>
		<b>Wireless Network Principles</b>	<ul style="list-style-type: none"> <li>✓ Explain Frequency Allocation and Regulation</li> <li>✓ Describe Antennas and Signal Propagation</li> <li>✓ Explain Multiplexing, Modulation and Media access control</li> <li>✓ Discuss the Classifications of Wireless Networks</li> </ul>
		<b>Wireless Local Area Networks, Cellular Networks, and Mobile IP</b>	<ul style="list-style-type: none"> <li>✓ Discuss the wireless network standards (i.e. IEEE802.11, 802.15,..)</li> <li>✓ Describe HiperLAN, Wireless Sensor Networks, and Zigbees</li> <li>✓ Explain the Principles of Cellular Networks</li> <li>✓ Discuss generations of Cellular Networks</li> <li>✓ Explain Mobile IP entities and Terminologies</li> <li>✓ Describe IP packet delivery, Registration, Tunneling, and Encapsulation</li> <li>✓ Explain Mobile ad-hoc networks</li> </ul>
		<b>Wireless Network Security</b>	<ul style="list-style-type: none"> <li>✓ Examine wireless LAN vulnerabilities</li> <li>✓ Discuss WLAN security models and policies</li> <li>✓ Describe how to secure wireless transmission using a VPN</li> </ul>
9	<b>Computer Security</b>	<b>Introduction to Computer Security</b>	<ul style="list-style-type: none"> <li>✓ Explain the terms, Threats, vulnerabilities, controls, risk</li> <li>✓ Describe security attacks, policies, mechanisms, preventions, detections, and deterrence</li> <li>✓ Discuss the software security assurance</li> </ul>
		<b>Computer Threat</b>	<ul style="list-style-type: none"> <li>✓ Mention and describe Malicious codes (i.e. viruses, Trojan horses, worms, spy-wares, etc.)</li> <li>✓ Explain the class of attacks and program flows</li> <li>✓ Explain how to controls to protect against program flaws in execution</li> </ul>

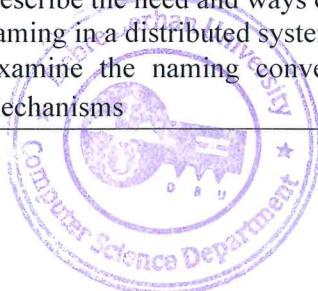


		<ul style="list-style-type: none"> <li>✓ Discuss the Database management systems security</li> </ul>
	<b>Cryptography and Encryption Techniques</b> <ul style="list-style-type: none"> <li>• Cipher Techniques</li> <li>• Conventional encryption algorithms</li> <li>• Cryptanalysis</li> <li>• Cryptographic Systems <ul style="list-style-type: none"> <li>◦ Symmetric key (DES, 3DES, AES, Block Cipher Modes)</li> <li>◦ Public key (Diffie-Hellman &amp; RSA)</li> </ul> </li> <li>• Digital Signature <ul style="list-style-type: none"> <li>◦ Using Public Key</li> <li>◦ Using Message Digest (i.e., MD4family, SHA family RIPEMD)</li> </ul> </li> <li>• Public-key Infrastructure (PKI)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Describe Basic cryptographic terms</li> <li>✓ Compare and contrast cipher techniques (i.e. Transposition &amp; Substitution)</li> <li>✓ Explain the basic requirements for trusted operating systems, and describe the independent evaluation, including evaluation criteria and evaluation process</li> <li>✓ Describe threats to networks, and explain techniques for ensuring network security, including encryption, authentication, firewalls, and intrusion detection</li> <li>✓ Mention and describe cryptographic systems</li> <li>✓ Discuss digital signature techniques</li> <li>✓ Explain public key infrastructure</li> <li>✓ Describe security requirements for database security, and describe techniques for ensuring database reliability and integrity, secrecy, inference control, and multi-level databases</li> </ul>
	<b>Network Security</b>	<ul style="list-style-type: none"> <li>✓ Explain the TCP/IP Suit Weaknesses and Buffer Overflows</li> <li>✓ Discuss Network security protocols and wireless security</li> </ul>
	<b>Security Mechanisms</b>	<ul style="list-style-type: none"> <li>✓ Mention and describe security mechanisms (i.e Firewall, Proxy server, IDS/IPS, VPN, etc.)</li> </ul>
	<b>Authentication and Access control</b>	<ul style="list-style-type: none"> <li>✓ Describe the basics of authentication and access controls</li> <li>✓ Mention and discuss types of Biometrics</li> <li>✓ Explain access control models (i.e., DAC, MAC, RBAC)</li> </ul>

		<b>Administering security</b>	<ul style="list-style-type: none"> <li>✓ Explain the requirements and techniques for security management, including security policies, risk analysis, and physical threats and controls</li> <li>✓ Describe cyber security</li> <li>✓ Explain the ethics of computing security</li> </ul>
10	<b>Network and System Administration</b>	<b>Introduction to System and Network Administration</b>	<ul style="list-style-type: none"> <li>✓ Explain the concepts, principles, and roles of system and network administration</li> <li>✓ Discuss the Role of the system and network administrator</li> <li>✓ Compare and contrast Unix-like Systems Vs Windows Systems</li> <li>✓ Describe the File system Hierarchy and Standard in Linux-like OS</li> <li>✓ Explain essential shell commands</li> </ul>
		<b>Account &amp; Security Administration, and Access Control</b>	<ul style="list-style-type: none"> <li>✓ Discuss account and user group management</li> <li>✓ Explain how to manage files and folder permission</li> <li>✓ Describe how to Manage Disk Quotas</li> </ul>
		<b>File Systems and Management of Data Storages</b>	<ul style="list-style-type: none"> <li>✓ Describe how to create, and mount file systems</li> <li>✓ Explain how to partition Disks with FDISK and Parted</li> <li>✓ Discuss how to configure Disk Quotas</li> </ul>
		<b>Network Management</b>	<ul style="list-style-type: none"> <li>✓ Discuss how to configure the Linux box, Web serve, DNS serve, proxy caches, and Postfix</li> <li>✓ Explain TCP/IP Troubleshooting: ping, traceroute, ifconfig, netstat, ipconfig</li> <li>✓ Describe how to Remote Administration with SSH and SCP</li> </ul>
		<b>Installation of Application Server and Management</b>	<ul style="list-style-type: none"> <li>✓ Compare and Contrast DHCP, DNS, and Telnet server with other Network OS setup of corresponding network services.</li> <li>✓ Discuss how to configure Open SSH, FTP, NFS, NIS, and Firewall</li> </ul>
		<b>Managing Network Services and Systems Security</b>	<ul style="list-style-type: none"> <li>✓ Describe how to Maintenance Troubleshooting: Common System and Network Problems</li> <li>✓ Discuss the General Strategies and how to Use Event Viewer and Troubleshoot Connectivity</li> <li>✓ Explain how to Resolve Boot Problems, Backup and Restore Data and System Volume</li> <li>✓ Explain Application Security and Login Security</li> </ul>



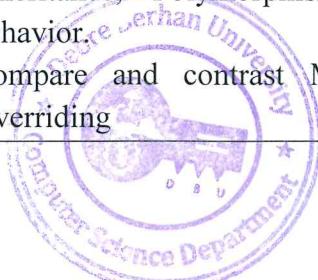
			<ul style="list-style-type: none"> <li>✓ Describe Boot Loader Security (LILO and GRUB) and Packet-Processing Model</li> </ul>
	<b>Windows Server Concepts and Administration</b>		<ul style="list-style-type: none"> <li>✓ Discuss how to select Computer systems to run the preferred windows server Domain Controller, Member Server, and/or Client</li> <li>✓ Explain and implement active directory roles and components</li> <li>✓ Create a LAN to connect Domain members</li> <li>✓ Install and configure Domain Controllers, Member Servers, and client computers</li> <li>✓ Create domain users, groups, and security policies</li> <li>✓ Join Member Servers and client computers in the domain</li> <li>✓ Arrange users, groups, and computers in organizational units</li> <li>✓ Implement and manage group policies</li> <li>✓ Implement Active Directory User Profiles and capabilities</li> <li>✓ Install, configure, access, and administer Windows Server services</li> </ul>
11	<b>Introduction to Distributed Systems</b>	<b>Introduction to Architecture of the distributed system</b>	<ul style="list-style-type: none"> <li>✓ Explain what a distributed system is, why they would design a system as a distributed system, and what the desired properties of such systems are.</li> <li>✓ Describe the problems and challenges associated with distributed systems, and evaluate the effectiveness and shortcomings of their solutions;</li> <li>✓ Examine the application of distributed system concepts in designing the recent large system</li> <li>✓ Describe different distributed algorithms over currently distributed platforms</li> <li>✓ Explain how to design and implement sample distributed systems.</li> </ul>
	<b>Process, Communication, Synchronization</b>		<ul style="list-style-type: none"> <li>✓ Explain and compare the communication mechanisms between different processes and systems.</li> <li>✓ Explain techniques apply to coordinate and synchronize multiple tasks in a distributed system.</li> <li>✓ Apply various distributed system algorithms related to clock synchronization, concurrency control, deadlock detection, etc.</li> </ul>
	<b>Naming</b>		<ul style="list-style-type: none"> <li>✓ Describe the need and ways of resources and entities naming in a distributed system</li> <li>✓ Examine the naming conventions and name resolution mechanisms</li> </ul>



		<b>Consistency and Replication</b>	✓ Reason about resource replication and how it improves performance and scalability in distributed systems. ✓ Examine algorithms that maintain consistent copies of replicas.
		<b>Fault Tolerance</b>	✓ Analyze fault tolerance and recovery in distributed systems and algorithms for the same ✓ Examine how to apply distributed system principles in ensuring transparency, consistency and fault tolerance
12	<b>Fundamentals of Database Systems</b>	<b>Basics of Database Systems</b> <ul style="list-style-type: none"><li>• Database and DBMS</li><li>• Roles in the database design environment (DBA, DBD user.)</li><li>• Types of Database Systems</li></ul> Database Languages (DDL, DML, DCL)	✓ Explain the following concepts: data, information, File, database, file system, and database system. ✓ Differentiate database and DBMS ✓ Explain the database system, and be able to identify its applications ✓ Identify the pros and cons of the manual approach, file-based approach and database approach ✓ Discuss the different roles in Database Design and Development ✓ Discuss the advantages and disadvantages of DBMS ✓ Explain the components of DBMS Languages
		<b>Database Modeling</b> <ul style="list-style-type: none"><li>• Relational, Hierarchical, and Network data model</li><li>• Relational Data Model</li><li>• Relational Integrity</li><li>• Key Constraints</li><li>• Relational DMS</li></ul>	✓ Explain database data models and schemas ✓ Discuss the different types of database data models ✓ Explain the different constraints (Integrity constraints, Domain constraints, Key constraints, Referential integrity constraint, Entity integrity constraint) ✓ Explain the operations performed in the relational data model (The Insert Operation, The Delete Operation, The Update Operation) ✓ Describe the advantages and disadvantages of using the Relational model
		<b>Database Design</b> <ul style="list-style-type: none"><li>• Conceptual<ul style="list-style-type: none"><li>◦ ER modeling (entity, attributes, Relationship)</li><li>◦ Enhanced ER model</li></ul></li><li>• Logical<ul style="list-style-type: none"><li>◦ Functional dependencies</li></ul></li></ul>	✓ Explain the purpose of normalization ✓ Illustrate the process of Normalization (1NF, 2NF, 3NF, BCNF) in a given database system ✓ Discuss Files of Unordered Records (Heap Files and Files of Ordered Records (Sorted Files)) ✓ Explain Hashing Techniques ✓ Discuss Index Structure for Files (Types of Single level Ordered Index, Dynamic Multilevel indexes using B- and Trees and B+ Trees, Indexes on Multiple Indexes)

	<ul style="list-style-type: none"> <li>○ Normalization (1NF, 2NF, 3NF)</li> <li>● Physical           <ul style="list-style-type: none"> <li>○ Physical database design processes</li> <li>○ DBMS file structure</li> </ul> </li> <li>● Indexing &amp; Hashing</li> </ul>	<ul style="list-style-type: none"> <li>✓ Illustrate how to design ER and EER Diagram</li> <li>✓ Discuss how to design different types of databases</li> </ul>
	<p><b>Query Language</b></p> <ul style="list-style-type: none"> <li>○ Relational Algebra</li> <li>○ Relational Calculus</li> <li>○ SQL</li> </ul> <p>Limitations of SQL</p>	<ul style="list-style-type: none"> <li>✓ Discuss the role of Relational Algebra in DBMS: Relational Algebra Operation Notations or Symbols, Advantages, and limitations of Relational Algebra</li> <li>✓ Explain the different database operations (Unary, binary)</li> <li>✓ Explain and apply the different SQL Languages (DML, DDL, DCL, TCL)</li> <li>✓ Apply the different SQL operations to a real problem.</li> <li>✓ Explain and use different SQL commands</li> </ul>
	<p><b>Advanced database concepts</b></p>	<ul style="list-style-type: none"> <li>✓ Explain information retrieval, data warehouse, data mining, and distributed database.</li> </ul>
13	<p><b>Advanced Database Systems</b></p>	<p><b>Concepts For Object-Oriented Databases</b></p> <ul style="list-style-type: none"> <li>✓ Describe Object Oriented Concepts (Abstraction, Encapsulation, and Information hiding)</li> <li>✓ Explain drawbacks of relational DBMS</li> <li>✓ Describe OO Database Design and Implementation</li> <li>✓ Describe data modeling and E-R diagramming</li> <li>✓ Discuss Object Identity, Object Structure, and Type Constructors</li> <li>✓ Explain type hierarchies and inheritance</li> </ul>
	<p><b>Query Processing and Optimization</b></p>	<ul style="list-style-type: none"> <li>✓ Explain database query processing and optimization</li> <li>✓ Translate SQL queries into relational algebra and then Query trees and graphs</li> <li>✓ Explain the basic concepts behind external sorting</li> <li>✓ Justify how to optimize SQL queries heuristically and cost estimation algorithms</li> </ul>
	<p><b>Transaction Management and Concurrency Control</b></p>	<ul style="list-style-type: none"> <li>✓ Explain the basics of transaction management</li> <li>✓ Discuss transaction schedules and serializability</li> <li>✓ Explain concurrency control</li> <li>✓ Discuss the different techniques of concurrency controls such as two-phase locking, timestamp ordering, and multi-version concurrency control</li> </ul>

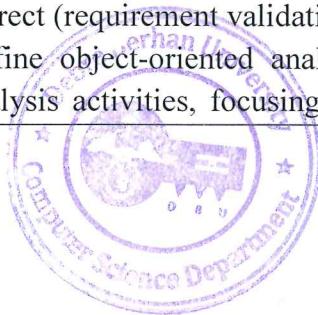
		<b>Database Integrity, Security, and Recovery Techniques</b>	<ul style="list-style-type: none"> <li>✓ Describe data security</li> <li>✓ Explain the basic database recovery techniques</li> <li>✓ Compare and contrast differed update and immediate update recovery techniques</li> <li>✓ Justify how to use different recovery methods when there is a database failure</li> </ul>
		<b>Distributed Database Systems</b>	<ul style="list-style-type: none"> <li>✓ Explain the concepts of the distributed database, types of DDB, and architectures of DDB.</li> <li>✓ Implement query processing, transaction management, and concurrency control techniques on distributed databases</li> <li>✓ Design a distributed database system in homogenous and heterogeneous environments</li> </ul>
		<b>Object-Oriented DBMS</b>	<ul style="list-style-type: none"> <li>✓ Explain object Oriented Concepts (Abstraction, Encapsulation, and Information hiding)</li> <li>✓ Discuss the drawbacks of relational DBMS</li> </ul>
		<b>Data warehousing and Data Mining Techniques</b>	<ul style="list-style-type: none"> <li>✓ Explain the concepts of data warehousing, data mining, and Online Transaction Processing (OLTP).</li> <li>✓ Discuss the different data mining techniques</li> <li>✓ Design and implement one complete DMS</li> </ul>
14	<b>Object-Oriented Programming</b>	<b>Introduction to Object-Oriented Programming</b>	<ul style="list-style-type: none"> <li>✓ Describe types of programming paradigms</li> <li>✓ Discuss the OO principles</li> <li>✓ Explain how to Edit, Compile, and Interpret OOP</li> </ul>
		<b>Class and Objects</b>	<ul style="list-style-type: none"> <li>✓ Define a call and create an object for OOP</li> <li>✓ Explain object-oriented design concepts, and have had practical experience in designing and implementing object-oriented software</li> <li>✓ Identify different types of methods, constructors, and instance members.</li> <li>✓ Distinguish types of access modifiers and scope of variables</li> <li>✓ Show how you can be able to write a class-based program for the real-world object</li> <li>✓ Explain how you can be able to create an object of class</li> </ul>
		<b>Object-Oriented Concepts</b> <ul style="list-style-type: none"> <li>• Encapsulation</li> <li>• Inheritance</li> <li>• Polymorphism</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain core concepts of OOP, such as Encapsulation, Inheritance, Polymorphism, Interface, and their behavior.</li> <li>✓ Compare and contrast Method Overloading and Overriding</li> </ul>



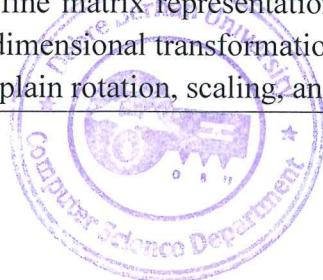
		<ul style="list-style-type: none"> <li>•Abstract classes and Interfaces</li> </ul>	<ul style="list-style-type: none"> <li>✓ Illustrate how to use special java keywords such as super, final, this, and instance-of for writing OOP</li> <li>✓ Analyze the concept of Abstraction, (in methods and classes)</li> </ul>
		<b>Exception Handling</b>	<ul style="list-style-type: none"> <li>✓ Differentiate various types of exception handling mechanisms (Try, catch, throw, and finally)</li> <li>✓ Describe how exceptions affect the design of an application)</li> <li>✓ Be able to handle exceptions</li> </ul>
		<b>Packages</b>	<ul style="list-style-type: none"> <li>✓ Describe packages in OOP</li> <li>✓ Explain CLASSPATH</li> <li>✓ Identify and apply package scope</li> </ul>
		<b>Data Structure</b>	<ul style="list-style-type: none"> <li>✓ Describe Data Structures in Java (sets, lists, queues, and maps)</li> <li>✓ Define map and dictionary in OOP</li> <li>✓ Write full programs by including basic OOP concepts</li> </ul>
15	<b>Java Programming</b>	<b>Overview of Java Programming</b>	<ul style="list-style-type: none"> <li>✓ Describe how a java program gets executed</li> <li>✓ Explain the basics of the java language</li> <li>✓ Manipulate data storage in the array</li> <li>✓ Illustrate the use of decision and repetition statements to solve a problem</li> </ul>
		<b>Java Applet Java GUI using JavaFX</b>	<ul style="list-style-type: none"> <li>✓ Describe the difference between java applet and java application</li> <li>✓ Identify Java GUI components of AWT, and Swing.</li> <li>✓ Explain how controls (Widgets) interact with a user</li> <li>✓ Identify properties of controls and Action/events handling methods</li> <li>✓ Explain the structure of JavaFX</li> <li>✓ Carry out design and development of complex UI elements,</li> <li>✓ Write a program that manipulates file and folder management</li> <li>✓ Explain how to create a GUI that can operate CRUD operations</li> <li>✓ Discuss how to write a GUI program including color, texts, fonts, lines, circles rectangles, and CSS styling</li> </ul>
		<b>Streams and File I/O</b>	<ul style="list-style-type: none"> <li>✓ Discuss the input-output streams</li> <li>✓ Explain I/O stream and file management</li> </ul>
		<b>Multi-threading concept</b>	<ul style="list-style-type: none"> <li>✓ Identify Threads and processes</li> <li>✓ Describe how multi-threading in java works</li> </ul>

			<ul style="list-style-type: none"> <li>✓ Create threads in two ways</li> <li>✓ Illustrate thread, multiple thread and process</li> <li>✓ Explain thread priorities and synchronization</li> <li>✓ Write a program that utilize a resource in parallel</li> </ul>
		<b>Networking in Java</b>	<ul style="list-style-type: none"> <li>✓ Illustrate how socket communication works</li> <li>✓ Write a program that communicate over a network in Client/Server environment using Sockets;</li> <li>✓ Describe how RMI works</li> <li>✓ Justify how to write a program that communicate with RMI</li> <li>✓ Write TCP/IP Client Server applications using Sockets</li> <li>✓ Write Java applications using the JDBC to make database independent queries; and Call methods remotely</li> </ul>
		<b>Java - Database Connectivity and Servlets</b>	<ul style="list-style-type: none"> <li>✓ Design a relational database application</li> <li>✓ Describe how a java application can be connected with a relational database system</li> <li>✓ Explain how to develop an application that uses a relational database system as a backend</li> <li>✓ Recall steps of DB Connection</li> <li>✓ Explain how a web application works</li> <li>✓ Describe a how a web request get processed</li> <li>✓ Describe what is servlets and how they will run</li> <li>✓ Write a web program using the java language</li> <li>✓ Call methods remotely, and develop web-based java application using servlet</li> <li>✓ Develop a desktop app in OO design and development using a convenient IDE.</li> <li>✓ Write Java applications using the JDBC to make database independent queries</li> </ul>
16	<b>Software Engineering</b>	<b>Object Oriented Concepts</b>	<ul style="list-style-type: none"> <li>✓ Explain the basics of system development life cycles</li> <li>✓ Define object (Attributes and operations) differentiate object with in a class</li> <li>✓ Explain the different (and complementary) ways to view software construction</li> <li>✓ Use the Object-Oriented system development methodology and understand techniques of modeling a system as a set of interacting or interrelated objects.</li> </ul>

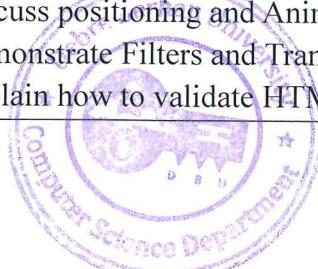
		<ul style="list-style-type: none"> <li>✓ Mention and explain the basics of object-oriented concepts</li> <li>✓ Explain the united approach (UA) of software development that combines the best practices, processes, and guidelines along with the UML</li> </ul>
	<b>UML</b>	<ul style="list-style-type: none"> <li>✓ Describe in detail, UML concepts, notations and methods concern to the Unified Modeling Language (UML)</li> <li>✓ Mention the advantage and disadvantage of object orientation for the new software paradigm</li> <li>✓ Explain the basic Building blocks of the UML</li> <li>✓ Analyze user requirements using UML of OO techniques.</li> <li>✓ Define the techniques of modeling aspects of systems</li> <li>✓ Create requirements using use case modeling concepts</li> </ul>
	<b>Requirements Elicitation</b>	<ul style="list-style-type: none"> <li>✓ Define functional and non-functional Requirements based on aspects of the system that are directly related to the system functionality or not</li> <li>✓ Explain the Criteria that are commonly used to evaluate good requirements</li> <li>✓ Enumerate Requirements elicitation activities</li> <li>✓ Analyze user requirements using UML of OO techniques</li> </ul>
	<b>Software Project Management</b>	<ul style="list-style-type: none"> <li>✓ Discuss the issues involved in software project management and the factors that affect</li> <li>✓ Explain responsibility of software project managers</li> <li>✓ Discuss the project size estimation metrics and techniques</li> <li>✓ Explain the SPMP document</li> <li>✓ Develop software project plans, supporting software quality plans and risk management plans</li> <li>✓ Perform a risk analysis on a project</li> <li>✓ Plan and Manage project risk</li> </ul>
	<b>Analysis</b>	<ul style="list-style-type: none"> <li>✓ Determining what to build in object-oriented analysis</li> <li>✓ List the mechanism for ensuring your requirement are correct (requirement validation techniques)</li> <li>✓ Define object-oriented analysis and enumerate the analysis activities, focusing on the identification of</li> </ul>



			<p>objects, behavior, relationships, classification, and their organization</p> <ul style="list-style-type: none"> <li>✓ Discuss the three basic individual models that are included with the analysis model</li> <li>✓ Explain the analysis activities that transform the use cases and scenarios produced during requirements elicitation into an analysis model</li> </ul>
		<b>Object Oriented System Design</b>	<ul style="list-style-type: none"> <li>✓ Determine the general transformation of an analysis model in to system design model</li> <li>✓ Determine how to build a system</li> <li>✓ Apply and analyzing design patterns effectively</li> <li>✓ Mention and describe the system design activities from object to subsystems</li> <li>✓ Understand system design from the design goal and software architecture perspective</li> <li>✓ Describe the basic system design concepts</li> </ul>
		<b>Software Quality Assurance</b>	<ul style="list-style-type: none"> <li>✓ Define software testing</li> <li>✓ Illustrate methods to implement software testing</li> <li>✓ Define software build</li> <li>✓ Explain software validation and software verification</li> <li>✓ Describe the software testing principles</li> </ul>
17	<b>Computer Graphics</b>	<b>Introduction to Computer Graphics</b>	<ul style="list-style-type: none"> <li>✓ Explain Computer Graphics and write down the application of computer graphics.</li> <li>✓ Discuss and list the components or elements of</li> <li>✓ Demonstrate the cathode Ray Tube</li> <li>✓ Discuss and differentiate the two-technique used in current computer graphics terminals for generating the image on the CRT screen</li> <li>✓ Discuss Graphics Hardware</li> </ul>
		<b>Introduction to the Rendering Process with OpenGL</b>	<ul style="list-style-type: none"> <li>✓ Describe the Role of OpenGL in the Reference Model</li> <li>✓ Explain how to use mathematical techniques to draw a lines</li> <li>✓ Identify and discuss the three-line drawing algorithms</li> <li>✓ Explain Output Primitives and Attributes</li> <li>✓ Draw a line and write its program using DDA</li> </ul>
		<b>Geometry, Line Generation, and Geometrical Transformations</b>	<ul style="list-style-type: none"> <li>✓ Describe points, lines, and polygons in OpenGL</li> <li>✓ Explain about 2-D transformation</li> <li>✓ Define matrix representation? Give an example of a 2-dimensional transformation</li> <li>✓ Explain rotation, scaling, and shearing with examples</li> </ul>



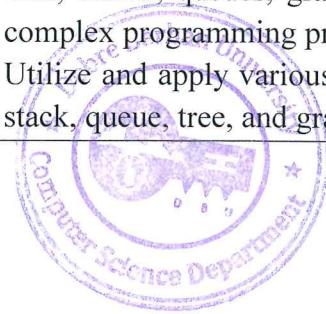
			<ul style="list-style-type: none"> <li>✓ Discuss different transformations; (i.e., composite, shear, 3D, and combination)</li> <li>✓ Demonstrate an example of 3-D transformation with matrix</li> <li>✓ Illustrate uniform and differential scaling</li> <li>✓ Explain basic state managements, normal vector and vector arrays</li> <li>✓ Demonstrate polygons and non-polygonal representation</li> </ul>
		<b>Application Modeling</b>	<ul style="list-style-type: none"> <li>✓ Compare and contrast image formats and their application</li> <li>✓ Justify how to be capable of using OpenGL to create interactive computer graphics</li> <li>✓ Discuss a typical graphics pipeline.</li> <li>✓ Explain storage strategies and matrix stacks</li> <li>✓ Explain how to make pictures on your computer</li> </ul>
18	<b>Web programming</b>	<b>Basics Of Internet</b>	<ul style="list-style-type: none"> <li>✓ Explain how Internet is working</li> <li>✓ Identify what are the required technologies to make web site available globally</li> <li>✓ Explain Web Hosting and Domain Name Registration</li> <li>✓ Be able to browse Internet</li> </ul>
		<b>Hyper Text Markup Language (HTML)</b>	<ul style="list-style-type: none"> <li>✓ Explain HTML</li> <li>✓ Identify HTML elements and their respective attributes</li> <li>✓ Develop web pages according to the business rule</li> <li>✓ Develop web site by considering accessibility, security and portability.</li> <li>✓ Be able to test a web page</li> </ul>
		<b>Cascading Style Sheets (CSS)</b> <ul style="list-style-type: none"> <li>• CSS Basics</li> <li>• Style Sheet Rules</li> <li>• Style Properties</li> <li>• CSS Measuring Units</li> </ul>	<ul style="list-style-type: none"> <li>✓ Describe the CSS</li> <li>✓ Be able to style web page contents [form, table, menu, layout, font, list, color] as required like inline, embedded or external CSS</li> <li>✓ Develop a web page responsive</li> <li>✓ Discuss the CSS Measuring Units</li> <li>✓ Be able to test the expected style</li> </ul>
		<b>Client-Side Scripting (JavaScript)</b>	<ul style="list-style-type: none"> <li>✓ Explain JavaScript</li> <li>✓ Manipulate HTML elements by JavaScript events</li> <li>✓ Discuss positioning and Animation</li> <li>✓ Demonstrate Filters and Transitions</li> <li>✓ Explain how to validate HTML form by JavaScript</li> </ul>



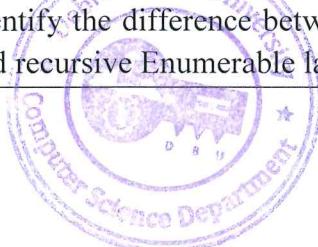
		<p><b>Server-Side Scripting Basic (PHP)</b></p> <p><b>File Database</b></p> <p><b>Cookies And Sessions</b></p> <p><b>Content Management Systems (CMS)</b></p>	<ul style="list-style-type: none"> <li>✓ Explain how server side scripting language works</li> <li>✓ Be able to make web content dynamic i.e., data manipulation to and from database and/or file</li> <li>✓ Discuss how to be able to secure the contents and test a program</li> <li>✓ Explain Cookies and Sessions</li> <li>✓ Describe the stateless model</li> <li>✓ Implement Cookies and Sessions in a website</li> <li>✓ Describe content management systems (CMS)</li> <li>✓ Develop website using CMS</li> </ul>
19	<b>Design and Analysis of Algorithms</b>	<b>Design and Analysis of Computer Algorithms</b>	<ul style="list-style-type: none"> <li>✓ Explain computational complexity analysis and various algorithm design paradigms</li> <li>✓ Illustrate algorithm analysis using the different techniques</li> <li>✓ Discuss how to analyze worst-case running times of algorithms using asymptotic analysis.</li> <li>✓ Discuss elementary data structures; linear data-structures (lists, array, linked list stack, queue and hashing table) and non-linear data structures (graph and tree)</li> </ul>
		<b>Algorithm Design Paradigm: [Divide and Conquer, Greedy Algorithms, Dynamic Programming, Branch &amp; Bound, Back Tacking, Parallel Algorithms]</b>	<ul style="list-style-type: none"> <li>✓ Describe the various design paradigms and explain why they are appropriate in an algorithmic design setting</li> <li>✓ Classify algorithms that employ a given design paradigm</li> <li>✓ Synthesize Divide and Conquer, Greedy, dynamic programming, branch &amp; bound, back tracking, parallel algorithms, and analyze them</li> <li>✓ Illustrate how to apply knowledge of computing and mathematics to algorithm design</li> <li>✓ Analyze a problem and identify the computing requirements appropriate for its solution</li> <li>✓ Demonstrate how to design, implement, and evaluate an algorithm to meet desired needs</li> <li>✓ Apply mathematical foundations, algorithmic principles, and computer science theory to the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices</li> </ul>



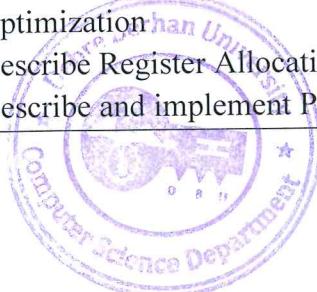
			<ul style="list-style-type: none"> <li>✓ Explain the design and development principles in the construction of software systems of varying complexity</li> <li>✓ Examine and apply the current techniques, skills, and tools necessary for computing practice</li> <li>✓ Discuss how to able to function effectively as a member of a team in order to accomplish a common goal and engage in continuing professional development</li> </ul>
20	<b>Data Structures and Algorithms</b>	<b>Data Structures and Algorithms</b>	<ul style="list-style-type: none"> <li>✓ Discuss the fundamental components and considerations in writing program; algorithm and data structure</li> <li>✓ Analyze the complexity of algorithms: write efficient program and analyze its efficiency in terms of time and space utilization</li> <li>✓ Explain and implement the notations that are used to describe a running time function of algorithms and utilize them</li> <li>✓ Compare and contrast the efficiency of algorithms or software solutions [best, average and worst-case scenario]</li> <li>✓ Elaborate how to use techniques of dynamic memory management method</li> <li>✓ Use mathematical techniques to analyses the efficiency of the various algorithms presented, as well as the common operations on the data structures, i.e., Big-Oh, Big-Omega, etc.</li> </ul>
		<b>Searching [Sequential, Binary and Hashing] and Sorting Algorithms [Insertion, Selection, Bubble and Shell, Heap, Quick and Merge]</b>	<ul style="list-style-type: none"> <li>✓ Explain searching and sorting algorithms</li> <li>✓ Discuss how to implement and use simple and advanced searching and sorting algorithms</li> <li>✓ Compare and select the best among several types of searching and sorting to apply in real-world problems</li> </ul>
		<b>Structures: Linked Lists, Stacks and Queues, Tree Structures and Graphs</b>	<ul style="list-style-type: none"> <li>✓ Describe and implement common data structures—lists, stacks, queues, graphs, and trees—for solving complex programming problems</li> <li>✓ Utilize and apply various data structures [linked list, stack, queue, tree, and graph] in real-world problems,</li> </ul>



			<ul style="list-style-type: none"> <li>✓ Apply the knowledge and skills of algorithms and data structures in various application domains</li> <li>✓ Discuss the application of binary trees</li> </ul>
21	<b>Automata and Complexity Theory</b>	<b>Model of Computation</b>	<ul style="list-style-type: none"> <li>✓ Define and understand different concept of automata theory and formal language</li> <li>✓ Define Alphabets, strings and language</li> <li>✓ Compare and manipulate formal description of language, automata and grammar</li> <li>✓ Design and understand Deterministic and Non-deterministic finite automata</li> <li>✓ Justify how to transform between equivalent non-deterministic finite automata to Deterministic finite automata</li> </ul>
		<b>Regular Expression and Regular Languages</b>	<ul style="list-style-type: none"> <li>✓ Design regular expression and regular grammar</li> <li>✓ Transform between regular language to regular expression and vice versa</li> <li>✓ Discuss how to apply Pumping lemma to prove the language is non-regular language</li> </ul>
		<b>Context Free Languages</b>	<ul style="list-style-type: none"> <li>✓ Define and manipulate different concept of Context free languages and Context free grammar</li> <li>✓ Construct Derivation tree or parse tree to a grammar</li> <li>✓ Define and show the ambiguity of a grammar</li> <li>✓ Create a simplified form of grammar</li> <li>✓ Classify grammar based on Chomsky's hierarchy of grammars</li> </ul>
		<b>Push Down Automata</b>	<ul style="list-style-type: none"> <li>✓ Discuss push down automata</li> <li>✓ Explain how to design non-deterministic pushdown automata and Deterministic push down automata</li> <li>✓ Identify the relation between Push down automata and context free languages</li> </ul>
		<b>Turing Machines</b>	<ul style="list-style-type: none"> <li>✓ Describe Turing machines performing simple task and for accepting or generating certain language</li> <li>✓ Identify Turing Decidable and Turing Acceptable language</li> <li>✓ Identify decidable and Undecidable problems</li> </ul>
		<b>Computability</b>	<ul style="list-style-type: none"> <li>✓ Describe computability theory</li> <li>✓ Explain Recursive functions in Computability theory</li> <li>✓ Identify the difference between Recursive languages and recursive Enumerable languages</li> </ul>



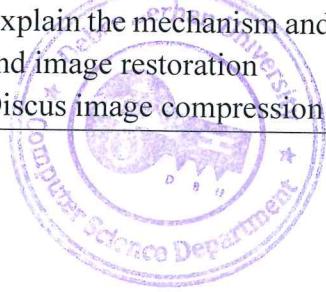
		<b>Computational Complexity</b>	<ul style="list-style-type: none"> <li>✓ Explain Computational complexity</li> <li>✓ Identify Complexity classes, P/NP, reduction and completeness and relation between Complexity classes</li> <li>✓ Discuss how to apply Cook's Theorem to different problems</li> </ul>
22	<b>Compiler Design</b>	<b>Introduction to Compiler Design</b>	<ul style="list-style-type: none"> <li>✓ Compare Compiler and Interpreter</li> <li>✓ Describe Compiler Design and the phases of Compiler Design</li> <li>✓ Identify the general computer language representation and implement the tools used in construction of compiler</li> </ul>
		<b>Lexical Analysis</b>	<ul style="list-style-type: none"> <li>✓ Compare Lexeme, Token and Pattern</li> <li>✓ Explain Automata, NFA, DFA and Conversion from NFA to DFA</li> <li>✓ Discuss Lexical Analyzer Generator tools to identify and recognize tokens</li> </ul>
		<b>Syntax Analysis</b>	<ul style="list-style-type: none"> <li>✓ Compare and contrast Lexeme, Token and Pattern</li> <li>✓ Discuss Automata, NFA, DFA and Conversion from NFA to DFA</li> <li>✓ Examine how to implement Lexical Analyzer Generator tools to identify and recognize tokens</li> </ul>
		<b>Syntax Directed Translation</b>	<ul style="list-style-type: none"> <li>✓ Describe Syntax Directed Definition and Construct Syntax Trees for a given language</li> <li>✓ Explain how to construct type checking rules and type conversion</li> <li>✓ Discuss Symbol Table using Hash Table</li> <li>✓ Describe Scope Management</li> </ul>
		<b>Intermediate Languages</b>	<ul style="list-style-type: none"> <li>✓ Compare and Contrast the Three Address Code methods (Quadruples, Triples, DAG, Tree and Indirect Triples)</li> <li>✓ Illustrate the Three Address Code methods</li> <li>✓ Identify the Instructions and Declarations in Three Address Code</li> </ul>
		<b>Code Generation and Optimization</b>	<ul style="list-style-type: none"> <li>✓ Identify the Goals and Techniques of Code Optimization</li> <li>✓ Describe Register Allocation</li> <li>✓ Describe and implement Peephole Optimization</li> </ul>



23	<b>Introduction to Artificial Intelligence (AI)</b>	<b>Introduction to AI</b> <ul style="list-style-type: none"> <li>• Goals of AI</li> <li>• Types of AI</li> <li>• Approaches to AI - making computer:           <ul style="list-style-type: none"> <li>◦ Think like a human</li> <li>◦ Act like a human</li> <li>◦ Think rationally</li> <li>◦ Act rationally</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>✓ Define the terms intelligence and Artificial Intelligence</li> <li>✓ Explain the different perspectives and historical background of AI.</li> <li>✓ Explain the goals of AI</li> <li>✓ Explain the limitations of current Artificial Intelligence techniques.</li> <li>✓ Discuss the different types of AI approaches</li> <li>✓ Mention classical examples of artificial intelligence</li> <li>✓ Discuss how to evaluate application of AI</li> </ul>
		<b>Intelligent Agents</b> <ul style="list-style-type: none"> <li>• Agents and Environments</li> <li>• Agent Types           <ul style="list-style-type: none"> <li>◦ Simple reflex, Model-based reflex, Goal-based, Utility-based and Learning agent</li> </ul> </li> <li>• Problem Solving (Goal Based) Agents</li> <li>• Learning Agent</li> </ul>	<ul style="list-style-type: none"> <li>✓ Define in your own words the following terms: agent, reflex agent, model-based agent, goal-based agent, utility-based agent and learning agent</li> <li>✓ Describe the different types and characteristics of intelligent agents</li> <li>✓ Properties of task environments</li> <li>✓ Differentiate the different types of searching strategies employed in goal-based agents</li> <li>✓ Compare and contrast the different searching algorithms</li> <li>✓ Describe dynamic game theory</li> </ul>
		<b>Knowledge Representation and Reasoning</b> <ul style="list-style-type: none"> <li>• Logical Agents</li> <li>• Propositional Logic</li> <li>• Predicate Logic</li> <li>• Reasoning under uncertainty</li> <li>• Planning</li> </ul>	<ul style="list-style-type: none"> <li>✓ Describe the reasoning, knowledge representation and learning techniques of artificial intelligence</li> <li>✓ Represent knowledge and implement inference techniques to provide solutions partially observable environments using propositional and first</li> <li>✓ Discuss the strengths and weaknesses of knowledge representation techniques and their applicability to different tasks</li> <li>✓ Explain propositional, First-Order, and Inference in First-Order logic</li> <li>✓ Discuss medical diagnosis</li> </ul>



		<b>Machine Learning Basics</b>	<ul style="list-style-type: none"> <li>✓ Discuss fundamental concepts in machine learning, the different classes of machine learning algorithms, and ways to choose and apply different basic machine learning algorithms</li> <li>✓ Explain how to evaluate the performance of learning systems</li> <li>✓ Compare and contrast Supervised and Unsupervised learning</li> <li>✓ Describe how to judge the suitability of a machine learning paradigm for a given problem and the available data</li> <li>✓ Discuss capabilities and limitations of the considered machine learning algorithms</li> <li>✓ Explain deep learning</li> </ul>
		<b>Natural Language Processing (NLP) Basics</b>	<ul style="list-style-type: none"> <li>✓ Explain fundamental concepts and ideas in natural language processing</li> <li>✓ Mention Machine learning Application in NLP</li> <li>✓ Discuss Sentiment Analysis, speech recognition, Chabot</li> </ul>
		<b>Robotic Sensing and Manipulation</b>	<ul style="list-style-type: none"> <li>✓ Explain sensing, manipulating in Robotics</li> <li>✓ Discuss human-robot interaction</li> <li>✓ Describe how autonomous robotic systems works</li> </ul>
		<b>Ethical and Legal Considerations in AI</b>	<ul style="list-style-type: none"> <li>✓ Explain privacy and Bias in AI</li> <li>✓ Discuss appropriate use of AI and future works</li> </ul>
24	<b>Computer Vision and Image Processing</b>	<b>Computer Vision and Image Processing</b>	<ul style="list-style-type: none"> <li>✓ Describe the fundamental concepts and the mathematical tools used in digital image processing and computer vision</li> <li>✓ Discuss the basic image processing steps to solve problems.</li> <li>✓ Explain applicability of image processing and computer vision</li> </ul>
		<b>Spatial &amp; Frequency Domain Image Processing</b>	<ul style="list-style-type: none"> <li>✓ Discuss how to perform image enhancement techniques in spatial and frequency domain.</li> <li>✓ Explain Fourier and different Transform and Its Inverse</li> </ul>
		<b>Image Restoration and Reconstruction, Image Compression</b>	<ul style="list-style-type: none"> <li>✓ Explain model of image and noise</li> <li>✓ Explain the elements of information theory</li> <li>✓ Explain the mechanism and types of data compression and image restoration</li> <li>✓ Discuss image compression Algorithms</li> </ul>



		<ul style="list-style-type: none"> <li>✓ Demonstrate image restoration, reconstruction and compression operations</li> <li>✓ Design and implement algorithms for advanced image analysis (e.g., image compression, image segmentation &amp; image representation)</li> </ul>	
	<b>Color Image Processing</b>	<ul style="list-style-type: none"> <li>✓ Explain color fundamentals, color models and full-Color Image Processing</li> <li>✓ Discuss morphological image processing and color transformation</li> </ul>	
	<b>Object Recognition</b>	<ul style="list-style-type: none"> <li>✓ Describe object detection and recognition techniques</li> <li>✓ Illustrate machine learning and deep learning for object recognition</li> <li>✓ Discuss deep learning for three-dimensional image analysis</li> <li>✓ Illustrate how to design and implement various algorithms for digital image processing and computer vision</li> </ul>	
25	<b>Research Methods in Computer Science</b>	<p><b>Introduction to Research</b></p> <ul style="list-style-type: none"> <li>• Scientific Research</li> <li>• Research Elements</li> <li>• Research Types and Approaches</li> <li>• Research Methods</li> <li>• Report Writing</li> </ul> <p><b>Formulation of The Research problem</b></p> <ul style="list-style-type: none"> <li>• Selection of a general topic</li> <li>• Literature review</li> <li>• Objectives and hypotheses</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explain research in Computer Science</li> <li>✓ Define and differentiate research and non-research issues</li> <li>✓ Explain scientific research activities in computing research</li> <li>✓ Discuss briefly important elements of research endeavors like objectives, motivations, significance, requirements, characteristics, types, approaches, methods, problem solving, report writing, results.</li> <li>✓ Describe the features of computing research and related activities</li> <li>✓ Identify relevant literature sources that have direct and indirect impact on the proposed research</li> <li>✓ Identify or select appropriate research topic or research problem</li> <li>✓ Formulate and evaluate a research hypothesis (research question).</li> <li>✓ Illustrate how to develop well organized research proposal based on the defined research topic</li> </ul>
	<b>Research process (Major Steps in the research process)</b>	<ul style="list-style-type: none"> <li>✓ Identify the overall process of designing a research study from its inception to its report.</li> <li>✓ Discuss the concepts and procedures of sampling, data collection, analysis and reporting.</li> </ul>	



	<ul style="list-style-type: none"> <li>• <b>Problem Definition</b></li> <li>• <b>Selection of Research Design</b> <ul style="list-style-type: none"> <li>○ Types of Research Designs</li> <li>○ Sampling Methods</li> <li>○ Data collection</li> </ul> </li> <li>• <b>Analysis, interpretation and presentation of data</b> <ul style="list-style-type: none"> <li>○ The Data Analysis Process</li> <li>○ kinds of data analysis</li> <li>○ Interpretation of data</li> <li>○ Report writing and presentation</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>✓ Identify appropriate research methodology (design principles, sampling strategies, data collection and analysis approaches)</li> <li>✓ Mention the basic stages of research design</li> <li>✓ Discuss the various sampling techniques and their use</li> <li>✓ Compare and contrast descriptive analysis and inferential analysis</li> <li>✓ Discuss how to organize and prepare technical papers, thesis and presentations</li> <li>✓ Elaborate how to write the actual research paper and defend the values of the results accordingly</li> <li>✓ Present insights and findings; written and oral reports</li> <li>✓ Conduct research effectively in IS/IT related fields that can solve real problems.</li> <li>✓ Work and cooperate effectively with other research workers on computing research</li> </ul>
	<p><b>Ethics in Research and project</b></p> <ul style="list-style-type: none"> <li>• Research ethics</li> <li>• Ethical and legal issues           <ul style="list-style-type: none"> <li>○ Plagiarism</li> <li>○ Falsification,</li> <li>○ Fabrication academic honesty related issues</li> </ul> </li> <li>• Data collection and analysis activities</li> </ul>	<ul style="list-style-type: none"> <li>✓ Discuss the research ethics and other related issues</li> <li>✓ Demonstrate ethical principles and code of conduct while doing scientific research, in general, and in the field of computing, in particular</li> <li>✓ Avoid issues related to plagiarism, falsification, fabrication academic dishonesty and other misconducts</li> <li>✓ Discuss how to implement proper data collection procedures in all stages of data gathering activities</li> <li>✓ Justify proper data analysis tools and techniques for effective research work</li> </ul>
26	<p><b>Selected Topics in Computer Science</b></p> <p><b>Topics Can Include:</b></p> <ul style="list-style-type: none"> <li>• Software Development Approach&amp; Software Architecture</li> </ul>	<ul style="list-style-type: none"> <li>✓ Explore the feature research topics in the area of Computer Science</li> </ul>

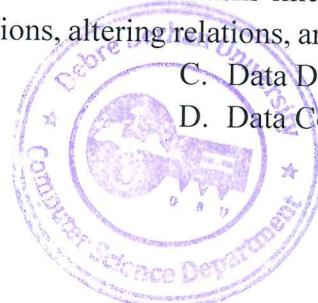


	<ul style="list-style-type: none"> <li>• Telecom &amp; Network Infrastructure and Technologies</li> <li>• Network &amp; Cyber Security</li> <li>• Storage and Database Technologies</li> <li>• Artificial Intelligence, Robotics, Drone Technology</li> <li>• Block-Chain Technology &amp; Cryptocurrency</li> <li>• Business Analysis and Big Data</li> <li>• Human Machine Interaction</li> <li>• Cloud, Edge, &amp; Pervasive Computing</li> <li>• Internet of Things</li> <li>• Smart Cities, Smart Living, ...</li> </ul>	<ul style="list-style-type: none"> <li>- System Administration</li> <li>- Database Administration</li> <li>- UI/UX Design</li> </ul> <p>✓ Describe the available and emerging Telecom and Network Infrastructure and technologies</p> <p>✓ Describe the Network and Cyber Security environment</p> <p>✓ Demonstrate the use of Artificial intelligence in different fields</p> <p>✓ Illustrate Block-chain Technology its advantage in cryptocurrency, banking, and other distribute system</p> <p>✓ Discuss the value of data and data big data analysis in business continuity</p> <p>✓ Describe and demonstrate Human machine interaction interface</p> <p>✓ Explain the concept of Cloud computing, types, advantages and disadvantages</p> <p>✓ Describe the concepts of Internet of Things, sensor networks, advantage in different fields</p> <p>✓ Illustrate the smart living and technologies toward smart living</p>
--	--	--

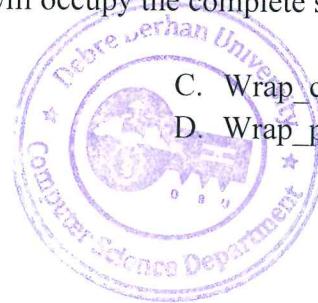


## **Sample Examination (the exam is prepared only from a few courses)**

1. BA collection of raw facts and figures that represent features or details about something is:
  - A. Information
  - B. Data
  - C. Knowledge
  - D. Wisdom
2. When we design a database model a functional dependency is the relation between
  - A. Entities
  - B. Attributes
  - C. Relations
  - D. Rows
3. When you omit parameters from a function call, values can be provided by
  - A. Formal parameters
  - B. Reference parameters
  - C. Overloaded parameters
  - D. Default parameters
4. Which of one the following packet delivery is used to send packets simultaneously to all stations over a network?
  - A. Broadcast
  - B. Multicast
  - C. Unicast
  - D. Single cast
5. In computer system, cache memory is used to:
  - A. Ensure fast booting
  - B. Replace the hard disk
  - C. Replace static memory
  - D. Speed-up memory access
6. A task of developing a technical blueprint and specifications for a solution that fulfills the business requirements is undertaken in the following phase of the system development process
  - A. System initiation
  - B. System design
  - C. System implementation
  - D. System analysis
7. In the relational data model, what is the formal name of the table?
  - A. Tuple
  - B. Relation
  - C. Rows
  - D. Attribute
8. The component of the CPU that translates machine instructions to the control signals (for themicro-operationss) is:
  - A. Registers
  - B. ALU
  - C. Control Unit
  - D. Accumulator (AC)
9. Among the following CSS selectors which one the used # tag?
  - A. Descendant selectors
  - B. ID selector
  - C. Class selector
  - D. Element selector
10. An 8-bit register contain binary value 10110110. What is the register value after following arithmetic shift right microoperations?
  - A. 01011011
  - B. 01101100
  - C. 11011011
  - D. 01101101
11. Which of the following is generally used for execution tasks like creating database and database objects of the relations, drop relations, altering relations, and truncating relations?
  - A. SQL Query
  - B. Data Manipulation Language
  - C. Data Definition Language
  - D. Data Control Language



12. Which one of the following CPU registers holds the address of the instructions (instructions in the program stored in memory) to be executed next?
- A. Address register
  - B. Instruction Register
  - C. Program Counter
  - D. Accumulator
13. Which one of the following is an error detection method?
- A. Multiplexing
  - B. Redundancy
  - C. Conditioning
  - D. Subnetting
14. Among the database models which type of database model is mainly used in the current day?
- A. Object-oriented database model
  - B. Network structure database model
  - C. Hierarchical Database model
  - D. Relational database model
15. Which android widget is used to display a text?
- A. Label
  - B. Text View
  - C. EditText
  - D. Text
16. In computer system, cache memory is used to:
- A. Ensure fast booting
  - B. Replace the hard disk
  - C. Replace static memory
  - D. Speed-up memory access
17. A database integrity that concerns correct values entered into the database is \_\_\_\_.
- A. Element integrity
  - B. Access control
  - C. Element accuracy
  - D. Recovery
18. During query processing, which operation consumes large resources relative to others?
- A. Selection
  - B. Projection
  - C. Join
  - D. Cross-product
19. Which of the following is considered as an element of cyber security?
- A. Network security
  - B. Software security
  - C. Application security
  - D. All of the above
20. Which of the following enables to access an HTML element from JavaScript?
- A. document.getElementById()
  - B. document.getElementsByTagName()
  - C. document.getElementsByName()
  - D. All of the above
21. Sharing of a single media for multiple communications is called?
- A. Multiplexing
  - B. DE Multiplexing
  - C. Multipoint communication
  - D. Full Duplex communication
22. The following xml layout attribute value will occupy the complete space available on the display of the device.
- A. Match\_parent
  - B. Fill\_content
  - C. Wrap\_content
  - D. Wrap\_parent



23. Given:

*grant select on User2.Table1 to User1 with grant option*

Which is true about the given query?

- A. User1 can select data from Table1.
- B. User1 cannot grant insert privilege to another user.
- C. User1 can delete data from Table1.
- D. User1 give the select privilege to User2.

24. In telephone network, digital devices are attached to the network via?

- |           |                |
|-----------|----------------|
| A. Switch | C. Modem       |
| B. Hub    | D. Multiplexer |

25. Which one of the following notations is used to represent two or more register transfer micro-operations that occur simultaneously?

- A. P: R3  $\neg$  R5; MAR  $\neg$  IR
- B. P: R3  $\neg$  R5= MAR  $\neg$  IR
- C. P, R3  $\neg$  R5 : MAR  $\neg$  IR
- D. P: R3  $\neg$  R5, MAR  $\neg$  IR

26. Among the following PHP built-in functions, which one enables to execute SQL queries?

- |                        |                      |
|------------------------|----------------------|
| A. mysqli_query()      | C. mysqli_connect    |
| B. Mysqli_fetch_rows() | D. mysqli_num_rows() |

27. Sending attack packets from multiple computers is known as

- |             |                      |
|-------------|----------------------|
| A. DDOS     | C. Masquerade        |
| B. Spoofing | D. Session hijacking |

28. Which dynamic mapping protocol helps the computer to get physical address for a given logical address?

- A. Reverse Address Resolution Protocol
- B. Internet Protocol
- C. Address Resolution Protocol
- D. Internet Control Message Protocol

29. Which is true about dirty bit?

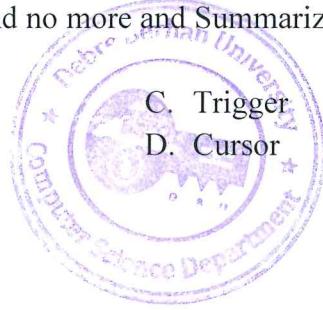
- A. Indicates whether a data item is locked
- B. Its value is 1, if a disk block is modified
- C. It is useless bit
- D. It has three bits

30. Assume, Ethio-Telecom granted a block of addresses to DASHEN BREWERY PLC at Debre Birhan cite, you know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

- A. 205.16.37.47
- B. 205.16.37.32
- C. 205.16.37.30
- D. 205.16.37.38



31. Consider  $DR \leftarrow M[AR]$ , Identify the operation.
- A. Memory Write
  - B. Memory Read
  - C. Arithmetic Operation
  - D. Shift Operation
32. In which one of the following concurrency control mechanism is read operation never rejected?
- A. Timestamp concurrency control
  - B. Two phase locking
  - C. Multi version concurrency control
  - D. Validation concurrency control
33. Given schedule s: r 1 (X); r 3 (X); w 1 (X); r 2 (X); w 3 (X). Which of the following pairs of operations results in total order schedule?
- A. r 1 (X); r 3 (X)
  - B. r 1 (X); w 1 (X)
  - C. r 2 (X); w 3 (X)
  - D. r 3 (X); w 1 (X)
34. Which of the following is the correct syntax to make heading 1 red and italic?
- A. `<h1 style ="color:red; font-style: italic;"> this heading will be red and italic</h1>`
  - B. `<style type="text/css">`  
`h1{ color:red;`  
`Font-style:italic;}`  
`</style>`
  - C. `<h1><font color="red"><i>this heading will be red and italic</i></font></h1>`
  - D. All
35. What is stored in the Stack Pointer?
- A. Operations
  - B. Addressing method
  - C. Stack data values
  - D. Address of top item
36. Which one of the following statements is incorrect?
- A. `$_POST[]` used to store form values which is sent by POST method
  - B. `setcookie()` enables to create cookie
  - C. `setsession()` used to start session
  - D. Data sent through POST method will not visible in the URL
37. In android, we can easily make a phone call from our android applications by invoking built-in phone calls app using Intents action
- A. ACTION\_CALL
  - B. ACTION\_VIEW
  - C. ACTION\_CONNECT
  - D. ACTION\_PHONE
38. Among the following which of the following is generally used for a composition of a table in the form of a predefined SQL query, can contain all rows of a table or select rows from a table, drive columns, restrict access to the data such that a user can see and sometimes modify exactly what they need and no more and Summarize data from various tables which can be used to generate reports.
- A. View
  - B. Stored procedure
  - C. Trigger
  - D. Cursor



39. In Classless Inter Domain Representation “/16” after an IP address means?

- A. The IP address belongs to class C
- B. In dotted decimal 255.255.254.253
- C. In binary 11111111 11111111 11110000 00000000
- D. 2 octets are reserved for network ID

40. Which one of the following statement is not correct?

- A. Repeater boosts or amplifies the signal before passing it through to the next section of cable.
- B. Multiplexing is a method of combining and transmitting several data streams across a single media.
- C. Straight through cabling are used to connect the same devices.
- D. Without Network interface card (NIC) a computer cannot be connected over a network.

41. What does the following line of code achieve?

```
Intent intent=new Intent  
(FirstActivity.this,SecondActivity.class);
```

- A. Create an implicit intent
- B. Create an explicit intent
- C. Start a service
- D. Start an activity

42. What is the application of computer graphics?

- A. Design processes particularly for engineering and architectural systems
- B. Education and training
- C. Image processing
- D. All can be answers

43. The proportional relationship between an image's width and height.

- A. Resolution
- B. Pixel
- C. Aspect ration
- D. Raster scan

44. A picture definition stored in a memory is called?

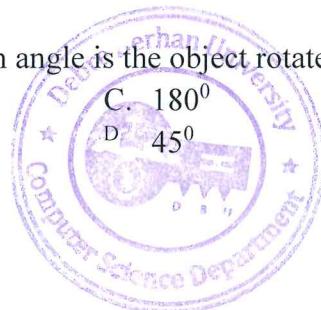
- A. Frame buffer or refresh buffer
- B. Picture element
- C. Resolution
- D. Horizontal retrace

45. An accurate and efficient raster line generating algorithm is called?

- A. Digital differential analyzer
- B. Bresenham's line algorithm
- C. Direct method
- D. All can be answers

46. In a reflection transformation by how much an angle is the object rotated?

- A.  $90^{\circ}$
- B.  $360^{\circ}$
- C.  $180^{\circ}$
- D.  $45^{\circ}$



47. In 8086 the overflow flag is set when \_\_\_\_\_.
- The sum is more than 16 bit
  - Carry and sign flags are set
  - Signed numbers go out of their range after an arithmetic operation
  - During subtraction
48. If there is a carry from lowest nibble during addition, \_\_\_\_\_ flag sets.
- |                    |              |
|--------------------|--------------|
| A. Carry           | C. Over flow |
| B. Auxiliary carry | D. Sign      |
49. If segment address = 1005 H, offset address = 5555 H, then the physical address is \_\_\_\_\_.
- |            |           |
|------------|-----------|
| A. 655A H  | C. 4550 H |
| B. 155A5 H | D. 56555  |
50. The alternate way of writing the instruction, ADD #5, R1 is \_\_\_\_\_
- |                  |                          |
|------------------|--------------------------|
| A. ADD [5],[R1]; | C. ADDIME 5,[R1];        |
| B. ADDI 5,R1;    | D. There is no other way |
51. If the 8085 adds 87H and 79 H, which of the following flags will become 1
- Zero flag and Auxiliary Carry flag
  - Zero flag and Carry flag
  - Carry flag and Auxiliary Carry flag
  - None of these
52. Which of the following is a condition that causes deadlock?
- |                     |                  |
|---------------------|------------------|
| A. Mutual exclusion | C. Circular wait |
| B. Hold and wait    | D. All of these  |
53. Which scheduler selects which processes should be brought into the ready queue?
- |              |               |
|--------------|---------------|
| A. Real-term | C. Mid-term   |
| B. Long-term | D. Short-term |
54. Fragmentation of the file system
- Occurs only if the file system is used improperly
  - Can always be prevented
  - Can be temporarily removed by compaction
  - is a characteristic of all file systems
55. The first step in systems Development Life Cycle is
- Database design
  - System design
  - Preliminary investigation and analysis
  - Graphical user interface
56. Which of the following is not a characteristic of a system?
- Operates for some purpose
  - Has homogeneous components
  - Has interacting components
  - Operates within a boundary



57. Which of the following statements is not true?
- An example of a hierarchical data structure is tree
  - Pointers are useful in traversing a linked list
  - A checksum digit is used for detecting Errors in data transmission
  - None of the above
58. Unit testing is
- Running the system with live data by the actual user
  - Making sure that the new programs do in fact process certain transactions according to Specifications
  - Is checking the logic of one or more programs in the candidate system
  - Testing changes made in an existing or a new program
59. A compiler is:
- A program that places program into memory and prepares them for execution
  - A program that automates the translation of assembly language into machine language
  - A program that accepts program written in high level language and produces an object program
  - A program that appears to execute a source program as if it were machine language
60. The concept of Finite State Automation is much used in this part of the compiler
- |                     |                      |
|---------------------|----------------------|
| A. lexical analysis | C. Semantic analysis |
| B. Syntax analysis  | D. Code optimization |
61. Bottom-up parser generates:
- Right most derivation
  - Right most derivation in reverse
  - Left most derivation
  - Left most derivation in reverse
62. Software engineers shall
- Act consistently with the public interest.”
  - Act in a manner that is in the best interests of his expertise and favour.”
  - Ensure that their products only meet the SRS.” d) all of the mentioned
  - All of the mentioned
63. Machine dependent phase of the compiler is
- Syntax analysis
  - Lexical analysis
  - Intermediate code generator
  - Code generator
64. In which of the three-address code does the instructions divided in to four fields:
- |               |                     |
|---------------|---------------------|
| A. Quadruples | C. Indirect triples |
| B. Triples    | D. DAG              |



65. What does the given CFG defines?  $S \rightarrow aSbS|bSaS| \epsilon$  and w denotes terminal  
A.  $ww^r$  C. Equal number of a's and b's  
B.  $wSw$  D. None of the mentioned

66. If the variables are suitably initialized, and if i remains within appropriate bounds, then the following code implements the stack operations Push and Pop when the stack is represented as an array  $V[1..N]$  with an index variable i.

Push: begin  $V[i] := x$  ;  $i := i + 1$  ; end

Pop: begin  $i := i - 1$ ;  $x := V[i]$  ; end

Which of the following gives the correct initialization for this stack implementation?

- A.  $i := 0$  D.  $i := N$   
B.  $i := 1$  E.  $i := N / 2$   
C.  $i := N - 1$
67. Which of the following C++ expressions does NOT always correctly compute the mathematical average of the integer variables a, b, c, and d?

- A. float  $((a + b + c + d) / 4.0)$   
B.  $(\text{float}(a + b + c + d)) / 4$   
C.  $(a + b + c + d) / 4$   
D.  $(a + b + c + d) / 4.0$   
E.  $(a + \text{float}(b) + c + d) / 4$

68. Consider the following recursive function.

```
int Fun ( int n )
{
    if ( n == 5 )
        return 2;
    else
        return 2 * Fun ( n + 1 );
}
```

What is the value returned by the function call  $\text{Fun} ( 2 )$  ?

- A. 2      B. 4      C. 8      D. 16      E. 24
69. Suppose I have the following recursive function, what is the return value when call  $\text{myop}(4)$ ?

```
int myop(int n){
if(n <= 2)
    return 1;
return 1 + myop(n-1) + myop(n-2)
}
```

- A. 5      B. 8      C. 7      D. 6



70. Which of the following is correct definition for Student structure which is used in LinkedList that stores student's info: id, gpa, name?

A.

```
struct Student{  
    int id;  
    double gpa;  
    string name;  
    Student* next;  
}
```

B.

```
structure Student{  
    int id;  
    double gpa;  
    string name;  
    Student* next;  
};
```

C.

```
struct Student{  
    int id;  
    double gpa;  
    string name;  
    Student* next;  
};
```

D.

```
struct Student{  
    int id;  
    double gpa;  
    string name;  
    Student next;  
};
```

71. If we use binary search algorithm to search 1 4 8 9 11 15 19 20 25 28 32 35 36 48 52 for key 17, list the elements that will be compared with key in order of when the comparison happens

A. 20 9 15 19

C. 20 9 15 17

B. 20 35 28 15

D. none of these

72. Apply the following insertion sort to array 4 1 2 5 3. Right after i is increased to 2, before the next iteration of loop body, what is the array?

```
void insertionSort(int arr[], int n)  
{  
    int i, key, j;  
    for (i = 1; i < n; i++)  
    {  
        key = arr[i];  
        j = i-1;  
        while (j >= 0 && arr[j] > key)  
        {  
            arr[j+1] = arr[j];  
            j = j-1;  
        }  
        arr[j+1] = key;  
    } }
```

A. 4 1 2 5 3

B. 1 4 2 5 3

C. 1 2 4 5 3

D. none of these



73. A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

- I. 81, 537, 102, 439, 285, 376, 305
- II. 52, 97, 121, 195, 242, 381, 472
- III. 142, 248, 520, 386, 345, 270, 307
- IV. 550, 149, 507, 395, 463, 402, 270

Suppose the BST has been unsuccessfully searched for key 273. Which all of the above sequences list nodes in the order in which we could have encountered them in the search? Answers:

- A. II and III
  - B. I and III
  - C. III and V
  - D. III only
74. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?
- A. 9 8 6 4 2 3 0 1 5 7
  - B. 7 4 1 0 3 2 4 6 8 9
  - C. 0 1 2 3 4 5 6 7 8 9
  - D. 0 2 4 3 1 6 5 9 8 7
75. What is the output of the following code segment?
- ```
char c = 'A';
System.out.println(c+32);
```
- A. 97
  - B. A
  - C. There is a compile error
  - D. c32
76. What is the output of the following Java code segment?
- ```
int arr[2];
System.out.println(arr[0]);
System.out.println(arr[1]);
```
- A. Compile error
  - B. 0 0
  - C. garbage value garbage value
  - D. Running time Exception
77. What is the output of the following Java code segment?
- ```
String s = "Hello"; // 1
System.out.println(s.size()); // 2
```
- A. Compile error in line // 1
  - B. Compile error in line // 2
  - C. 5
  - D. 6
78. Which among the following looks similar to the given expression?  
((0+1)(0+1)) \*

- A.  $\{x \in \{0,1\}^* | x \text{ is all binary number with even length}\}$
- B.  $\{x \in \{0,1\}^* | x \text{ is all binary number with even length}\}$
- C.  $\{x \in \{0,1\}^* | x \text{ is all binary number with odd length}\}$
- D.  $\{x \in \{0,1\}^* | x \text{ is all binary number with odd length}\}$



79. A grammar  $G = (V, T, S, P)$  is said to be context free if all productions are the form  $x \rightarrow y$ , where
- A.  $x, y \in (VuT)^+$  and  $|x| \geq |y|$
  - B.  $x, y \in (VuT)^*$  and  $|x| \leq |y|$
  - C.  $x, y \in (VuT)^+$  and  $|x| \leq |y|$
  - D. None of the above
80. What is the minimum number of states required by a Turing machine to accept a string which consists of even number of 1's?
- A. 5
  - B. 3
  - C. 1
  - D. 2
81. Which one of the following sequences always guarantees that the removal of unwanted variables and productions when simplifying a CFG?
- A. Removing Null-able Variables, Unit-Productions, and Useless Variables
  - B. Removing Useless Variables, Null-able Variables, and Unit-Productions
  - C. Removing Unit-Productions, Null-able Variables, and Useless Variables
  - D. None of the above
82. Which one of the following problems is decidable?
- A. Halting problem for Turing machine
  - B. Whether a CFG generates all the string or not
  - C. Convert the Ambiguous of CFG to non-Ambiguous CFG
  - D. Membership problem of finite state machine M to a given string W
83. The goal of requirement engineering is to develop and maintain sophisticated and descriptive \_\_\_\_\_ document.
- A. Feasibility Study
  - B. Requirement Gathering
  - C. Software Requirement Validation
  - D. System Requirements Specification
84. Which one of the following models is not suitable for accommodating any change?
- A. Prototyping Model
  - B. Waterfall Model
  - C. RAD Model
  - D. Build & Fix Model
85. The process to gather the software requirements from client, analyze and document them is known as \_\_\_\_\_
- A. Feasibility Study
  - B. Requirement Gathering
  - C. Requirement Engineering
  - D. System Requirements Specification



## Answer Key

- |       |       |       |       |       |
|-------|-------|-------|-------|-------|
| 1. B  | 18. D | 35. D | 52. D | 69. A |
| 2. B  | 19. D | 36. C | 53. B | 70. C |
| 3. D  | 20. D | 37. A | 54. C | 71. A |
| 4. A  | 21. A | 38. A | 55. C | 72. B |
| 5. D  | 22. A | 39. D | 56. B | 73. B |
| 6. B  | 23. A | 40. C | 57. B | 74. C |
| 7. C  | 24. C | 41. D | 58. D | 75. A |
| 8. C  | 25. D | 42. D | 59. C | 76. A |
| 9. B  | 26. A | 43. C | 60. A | 77. B |
| 10. C | 27. A | 44. A | 61. B | 78. A |
| 11. C | 28. C | 45. B | 62. A | 79. D |
| 12. C | 29. B | 46. C | 63. D | 80. D |
| 13. B | 30. B | 47. C | 64. A | 81. A |
| 14. D | 31. B | 48. B | 65. C | 82. D |
| 15. B | 32. C | 49. B | 66. B | 83. D |
| 16. D | 33. B | 50. B | 67. C | 84. B |
| 17. A | 34. D | 51. C | 68. D | 85. C |

