

**SCHEME OF TEACHING AND EXAMINATION  
COMPUTER SCIENCE AND ENGINEERING  
III SEMESTER**

Sl. No.	Subject Code	Subject	Teaching Dept.	Teaching hours / week		Examination Duration & Marks			
				Th	Pr.	Hrs.	I.A.	Exam	Total
1.		Maths III	Maths	4	-	3	25	100	125
2.	06CSE/06ISE	Electronic Circuits	CSE/ISE	4	-	3	25	100	125
3	06CSE/06ISE	Logic Design	CSE/ISE	4	-	3	25	100	125
4	06CSE/06ISE	Discrete Mathematical Structures	CSE/ISE	4	-	3	25	100	125
5	06CSE/06ISE	Data Structures with C	CSE/ISE	4	-	3	25	100	125
6	06CSE/06ISE	Unix and Shell Programming	CSE/ISE	4	-	3	25	100	125
7	06CSEL/06ISEL	Data Structures Lab	CSE/ISE	-	3	3	25	50	75
8	06CSEL/06ISEL	Electronic Circuits & Logic Design Lab.	CSE/ISE	-	3	3	25	50	75
<b>Total</b>				<b>24</b>	<b>6</b>	<b>-</b>	<b>200</b>	<b>700</b>	<b>900</b>

COMPUTER SCIENCE & ENGINEERING

**SCHEME OF TEACHING AND EXAMINATION  
COMPUTER SCIENCE AND ENGINEERING  
IV SEMESTER**

Sl. No.	Subject Code	Subject	Teaching Dept.	Teaching hours / week		Examination Duration & Marks			
				Th.	Pr.	Hrs.	I.A.	Exam	Total
1.		Maths IV	Maths	4	-	3	25	100	125
2.	06CSE/06ISE	Graph Theory and Combinatorics	CSE/ISE	4	-	3	25	100	125
3	06CSE/06ISE	Analysis and Design of Algorithms	CSE/ISE	4	-	3	25	100	125
4	06CSE/06ISE	Object Oriented Programming	CSE/ISE	4	-	3	25	100	125
5	06CSE/06ISE	Microprocessors	CSE/ISE	4	-	3	25	100	125
6	06CSE/06ISE	Computer Organization	CSE/ISE	4	-	3	25	100	125
7	06CSEL/06ISEL	Object Oriented Programming Lab.	CSE/ISE	-	3	3	25	50	75
8	06CSEL/06ISEL	Microprocessors Lab.	CSE/ISE	-	3	3	25	50	75
<b>Total</b>				<b>24</b>	<b>6</b>	<b>-</b>	<b>200</b>	<b>700</b>	<b>900</b>

**Electronic Circuits**  
(Common to CSE & ISE)

<b>Sub Code</b>	<b>: 06CSE/06ISE32</b>	<b>IA Marks</b>	<b>: 25</b>
<b>Hrs / Week</b>	<b>: 04</b>	<b>Exam Hours</b>	<b>: 03</b>
<b>Total Hrs</b>	<b>: 52</b>	<b>Exam Marks</b>	<b>: 100</b>

Unit I

1. Diode Circuits: Clippers and Limiters, Clampers
2. Special-Purpose Devices: Optoelectronic Devices, The Schottky Diode, The Varactor, Other Diodes.

**6 Hours**

Unit II

3. Transistor AC Models: Base-Biased Amplifier, Emitter-Biased Amplifier, Small-Signal Operation, AC Beta, AC Resistance of the Emitter Diode, Two Transistor Models, Analyzing an Amplifier, AC Quantities on the Data Sheet

**6 Hours**

Unit III

4. Voltage Amplifiers: Voltage Gain, The Loading Effect of Input Impedance, Multistage Amplifiers, Swamped Amplifier, Two-Stage Feedback, Troubleshooting
5. CC and CB Amplifiers: CC Amplifier, Output Impedance, Cascading CE and CC, Darlington Connections, Voltage Regulation, The Common-Base Amplifier

**7 Hours**

Unit IV

6. Power Amplifiers: Amplifier Terms, Two Load Lines, Class A Operation, Class B Operation, Class B Push-Pull Emitter Follower, Biasing Class B/AB Amplifiers, Class B/AB Driver, Class C Operation, Class C Formulas, Transistor Power Rating

**7 Hours**

Unit V

7. MOSFETs: The Depletion-Mode MOSFET, D-MOSFET Curves, Depletion-Mode MOSFET Amplifiers, The Enhancement-Mode MOSFET, The Ohmic Region, Digital Switching, CMOS

**6 Hours**

Unit VI

8. Frequency Effects: Frequency Response of an Amplifier, Decibel Power Gain, Decibel Voltage Gain, Impedance Matching, Decibels above a Reference, Bode Plots

9. Negative Feedback: Four Types of Negative Feedback, VCVS Voltage Gain, Other VCVS Equations, The ICVS Amplifier, The VCIS Amplifier, The ICIS Amplifier, Bandwidth

**7 Hours**

#### Unit VII

10. Nonlinear Op-Amp Circuits: Comparators with Zero Reference, Comparators with Nonzero References, Comparators with Hysteresis, Window Comparator, Integrator, Waveform Conversion, Waveform Generation

11. Oscillators: The 555 Timer, Astable Operation of the 555 Timer, 555 Circuits, The Phase-Locked Loop, Function Generator ICs

**7 Hours**

#### Unit VIII

12. Regulated Power Supplies: Supply Characteristics, Shunt Regulators, Series Regulators, Monolithic Linear Regulators, Current Boosters, DC-DC Converters, Switching Regulators

**6 Hours**

#### Text Book

1. Electronic Principles, Albert Malvino & David J Bates, 7<sup>th</sup> Edition, TMH, (2007)  
(Chapter 4-10, 4-11, Chapter 5-8,5-9,5-10,5-11, Chapters 9, 10, 11, 12, Chapter 14-1,14-2, 14-3, 14-4, 14-5, 14-6, 14-7, Chapter 16-1, 16-2, 16-3, 16-4, 16-5, 16-6, Chapter 19, Chapter 22-1, 22-2, 22-3, 22-4, 22-5, 22-6, 22-7, Chapter 23-7, 23-8, 23-9, 23-10, 23-11 and Chapter 24)

#### Reference Books

1. Electronic Devices and Circuit Theory, Robert L. Boylestad, Louis Nashelsky, 9<sup>th</sup> Edition, PHI/Pearson Education, 2006
2. Electronic Devices and Circuits, David A. Bell, 4<sup>th</sup> Edition, PHI, 2006

**Logic Design**  
(Common to CSE & ISE)

<b>Sub Code</b>	<b>: 06CSE/06ISE 33</b>	<b>IA Marks</b>	<b>25</b>
<b>Hrs / Week</b>	<b>: 04</b>	<b>Exam Hours</b>	<b>: 03</b>
<b>Total Hrs</b>	<b>: 52</b>	<b>Exam Marks</b>	<b>: 100</b>

Unit I

1. Digital Logic: Overview of Basic Gates and Universal Logic Gates, AND-OR-Invert Gates, Positive and Negative Logic, Introduction to HDL

**2 Hours**

2. Combinational Logic Circuits: Boolean Laws and Theorems, Sum-of-products Method, Truth Table to Karnaugh Map, Pairs, Quads, and Octets, Karnaugh Simplifications, Don't Care Conditions, Product-of-sums Method, Product-of-sums Simplification, Simplification by Quine-McClusky Method, Hazards and Hazard Covers, HDL Implementation Models

**5 Hours**

Unit II

3. Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD-to-Decimal Decoders, Seven-segment Decoders, Encoders, EX-OR gates, Parity Generators and Checkers, Magnitude Comparator, Read-only memory, Programmable Array Logic, Programmable Logic, Troubleshooting with a Logic Probe, HDL Implementation of Data Processing Circuits

**7 Hours**

Unit III

4. Arithmetic Circuits: Binary Addition, Binary Subtraction, Unsigned Binary Numbers, Sign-Magnitude Numbers, 2's Complement Representation, 2's Complement Arithmetic, Arithmetic Building Blocks, The Adder-Subtractor, Fast Adder, Arithmetic Logic Unit, Binary Multiplication and Division, Arithmetic Circuits using HDL

**6 Hours**

Unit IV

5. Clocks and Timing Circuits: Clock Waveforms, TTL Clock, Schmitt Trigger, Monostables with Input Logic, Pulse-forming Circuits

**2 Hours**

6. Flip-Flops: RS Flip-flops, Gated Flip-flops, Edge-triggered RS, D, JK Flip-flops, Flip-flop timing, JK Master-slave Flip-flops, Switch Contact Bounce Circuits, Various Representations of Flip-flops, Analysis of Sequential Circuits, Conversion of Flip-flops – a synthesis example, HDL implementation of Flip-flop

**5 Hours**

#### Unit V

7. Registers: Types of Registers, Serial In-Serial Out, Serial In-Parallel Out, Parallel In-Serial Out, Parallel In-Parallel Out Applications of Shift Registers, Register Implementation in HDL

**2 Hours**

8. Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus, Decade Counters, Presettable Counters, Counter Design as a Synthesis Problem, A Digital Clock, Counter Design Using HDL

**5 Hours**

#### Unit VI

9. Design of Sequential Circuit: Model Selection, State Transition Diagram, State Synthesis Table, Design Equations and Circuit Diagram, Implementation using Read Only Memory, Algorithmic State Machine, State Reduction Technique, Analysis of Asynchronous Sequential Circuit, Problems with Asynchronous Sequential Circuits, Design of Asynchronous Sequential Circuit

**6 Hours**

#### Unit VII

10. D/A Conversion and A/D Conversion: Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, Continuous A/D Conversion, A/D Techniques, Dual-Slope A/D Conversion, A/D Accuracy and Resolution

**6 Hours**

#### Unit VIII

11. Digital Integrated Circuits: Switching Circuits, 7400 TTL, TTL Parameters, TTL Overview, Open-collector Gates, Three-state TTL Devices, External Drive for TTL Loads, TTL Driving External Loads, 74C00 CMOS, CMOS Characteristics, TTL-to-CMOS Interface, CMOS-to TTL Interface

**6 Hours**

## Text Book

1. Digital Principles and Applications, Donald P Leach, Albert Paul Malvino & Goutam Saha, 6<sup>th</sup> Edition, TMH, 2006  
(Chapters 2, 3, 4, 6, Chapter 7-1, 7-2, 7-3, 7-6, 7-7, Chapters 8, 9, 10, 11-1 to 10, 12, 14-1 to 12)

## Reference Books

1. Fundamentals of Digital Logic with Verilog Design, Stephen Brown, Zvonko Vranesic, TMH, 2006
2. Fundamentals of Logic Design, Charles H. Roth, Jr., 5<sup>th</sup> Edition, Thomson, 2004
3. Digital Systems Principles and Applications, Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, 10<sup>th</sup> Edition, PHI Pearson Education, 2007

### Discrete Mathematical Structures

(Common to CSE & ESE)

Sub Code	: 06CSE/06ISE 14	IA Marks	: 25
Hrs / Week	: 04	Exam Hours	: 03
Total Hrs	: 52	Exam Marks	: 100

#### Unit I

1. Set Theory: Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams, A First Word on Probability, Countable and Uncountable Sets

**6 Hours**

#### Unit II

2. Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference

**7 Hours**

#### Unit III

3. Fundamentals of Logic *contd.*: The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems

**6 Hours**

#### Unit IV

4. Properties of the Integers: Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions

**6 Hours**

#### Unit V

5. Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions – Stirling Numbers of the Second Kind, Special Functions, The Pigeon-hole Principle, Function Composition and Inverse Functions

**7 Hours**

#### Unit VI

6. Relations *contd.*: Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions

**7 Hours**

#### Unit VII

7. Groups: Definitions, Examples, and Elementary Properties, Homomorphisms, Isomorphisms, and Cyclic Groups, Cosets, and Lagrange's Theorem
8. Coding Theory and Rings: Elements of Coding Theory, The Hamming Metric, The Parity Check, and Generator Matrices

**6 Hours**

#### Unit VIII

9. Group Codes: Decoding with Coset Leaders, Hamming Matrices
10. Rings and Modular Arithmetic: The Ring Structure – Definition and Examples, Ring Properties and Substructures, The Integers Modulo  $n$

**7 Hours**

#### Text Book

1. Discrete and Combinatorial Mathematics, Ralph P. Grimaldi, 5<sup>th</sup> Edition, PHI Pearson Education, 2004  
(Chapter 3.1, 3.2, 3.3, 3.4, Appendix 3, Chapter 2, Chapter 4.1, 4.2, Chapter 5.1 to 5.6, Chapter 7.1 to 7.4, Chapter 16.1, 16.2, 16.3, 16.5 to 16.9, and Chapter 14.1, 14.2, 14.3)

#### Reference Books

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, 6<sup>th</sup> Edition, McGraw Hill, 2007
2. Discrete Mathematical Structures: Theory and Applications, D.S. Malik and M.K. Sen, Thomson, 2004
3. Discrete Mathematics with Applications, Thomas Koshy, Elsevier, 2005
4. A Treatise on Discrete Mathematical Structures, Jayant Ganguly, Sanguine Technical Publishers, 2006

**Data Structures with C**  
(Common to CSE & ISE)

<b>Sub Code</b>	<b>: 06CSE/06ISE 35</b>	<b>IA Marks</b>	<b>: 25</b>
<b>Hrs / Week</b>	<b>: 04</b>	<b>Exam Hours</b>	<b>: 03</b>
<b>Total Hrs</b>	<b>: 52</b>	<b>Exam Marks</b>	<b>: 100</b>

**Part A**

C Language Features

Unit I

1. Pointers: Concepts, Pointer variables, Accessing variables through pointers, Pointer declaration and definition, Initialization of pointer variables, Pointers and functions, Pointer to pointers, Compatibility, Lvalue and Rvalue, Arrays and pointers, Pointer arithmetic and arrays, Passing an array to a function, Understanding complex declarations, Memory allocation functions, Array of pointers.

**7 Hours**

Unit II

2. Strings: String concepts, C strings, String I/O functions, Array of strings, String manipulation function, Memory formatting.

**2 Hours**

3. Derived types -Enumerated, Structure, and Union: The type definition, Enumerated types, Structure, Accessing structures, Complex structures, Array of structures, Structures and functions, Unions

**3 Hours**

4. Binary Files: Classification of Files, Using Binary Files, Standard Library Functions for Files

**2 Hours**

**Part B**

Unit III

5. The Stack: Definition and Examples, Representing Stacks in C, An Example – Infix, Postfix, and Prefix

**7 Hours**

Unit IV

6. Recursion: Recursive Definition and Processes, Recursion in C, Writing Recursive Programs, Simulating Recursion, Efficiency of Recursion

**4 Hours**

7. Queues: The Queue and its Sequential Representation

**2 Hours**

Unit V

8. Lists: Linked Lists, Lists in C, An Example – Simulation using Linked Lists

**6 Hours**

**Unit VI**

1. Lists *contd.*: Other List Structures

**6 Hours**

**Unit VII**

10. Trees: Binary Trees, Binary Tree Representations

**6 Hours**

**Unit VIII**

11. Trees *contd.*: Representing Lists as Binary Trees, Trees and their applications

**7 Hours**

**Text Books**

1. Behrouz A. Forouzan and Richard F. Gilberg, Computer Science A Structured Programming Approach Using C, Second Edition, Thomson, 2003  
(Chapter 9.1 to 9.9, Chapter 10.1 to 10.6, Chapter 11.1 to 11.6, Chapter 12.1 to 12.8, Chapter 13.1 to 13.3)
2. Data Structure using C, Aaron M. Yenenbaum, Yedidyah Langsam & Moshe J. Augenstein, Pearson Education/PHI, 2006  
(Chapter 2, 3, 4, 5.1, 5.2, 5.4, 5.5)

**Reference Books**

3. Richard F. Gilberg and Behrouz A. Forouzan, Data Structures A Pseudocode approach with C, Thomson, 2005
4. Robert Kruse & Bruce Leung, Data Structures & Program Design in C, Pearson Education, 2007

**Note**

The Question paper consists of two parts A and B containing 2 and 6 questions respectively. The student is required to answer any 5 questions selecting at least one question from part A.

**UNIX and Shell Programming**

*(Common to CSE & ISE)*

**Sub Code : 06CSE/06ISE 36**  
**Hrs / Week : 04**

**IA Marks : 25**  
**Exam Hours : 03**

**Total Hrs : 52**

**Exam Marks : 100**

**Unit I**

1. The Unix Operating System, The UNIX architecture and Command Usage, The File System

**6 Hours**

**Unit II**

2. Basic File Attributes, The vi Editor

**6 Hours**

**Unit III**

3. The Shell, The Process, Customizing the environment

**7 Hours**

**Unit IV**

4. More file attributes, Simple filters

**6 Hours**

**Unit V**

5. Filters using regular expressions,

**6 Hours**

**Unit VI**

6. Essential Shell Programming

**7 Hours**

**Unit VII**

5. awk – An Advanced Filter

**7 Hours**

**Unit VIII**

- perl - The Master Manipulator

**7 Hours**

**Text Book**

1. Sumitabha Das, “UNIX – Concepts and Applications”, 4<sup>th</sup> Edition, Pearson McGraw Hill, 2006  
(Chapters 1.2, 2, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 18, 19)

**Reference Books**

1. Behrouz A. Forouzan and Richard F. Gilberg, UNIX and Shell Programming, Thomson, 2005
2. Unix & Shell Programming, M.G. Venkateshmurthy, Pearson Education, 2005

**Data Structures Laboratory**

*(Common to CSE & ISE)*

<b>Sub Code</b>	<b>: 06CSEL/06ISEL 37</b>	<b>IA Marks</b>	<b>: 25</b>
<b>Hrs / Week</b>	<b>: 03</b>	<b>Exam Hours</b>	<b>: 03</b>
<b>Total Hrs</b>	<b>: 42</b>	<b>Exam Marks</b>	<b>: 50</b>

1. Write a C Program to create a sequential file with atleast 5 records, each record having the structure shown below:

	Name	Marks1	Marks2	Marks3
Non-zero positive integer	25 Characters	Positive Integer	Positive Integer	Positive Integer

Write necessary functions

- To display all the records in the file.
- To search for a specific record based on the USN. In case the record is not found, suitable message should be displayed. Both the options in this case must be demonstrated.

2. Write and demonstrate the following C functions:

- newStrCpy that does the same job as strcpy
- newStrCat that does the same job as strcat without using any library functions.

3. Write a C Program which accepts the Internet Protocol (IP) address in decimal dot format (ex. 153.18.8.105) and converts it into 32-bit long integer (ex. 2568095849) using strtok library function and unions.

4. Write a C program to construct a stack of integers and to perform the following operations on it:

- Push
- Pop
- Display

The program should print appropriate messages for stack overflow, stack underflow, and stack empty.

5. Write a C Program to convert and print a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), \* (multiply) and / (divide).

6. Write a C Program to evaluate a valid suffix/postfix expression using stack. Assume that the suffix/postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), \* (multiply) and / (divide).

7. Write a C Program to simulate the working of a queue of integers using an array. Provide the following operations:

- a. Insert
- b. Delete
- c. Display

8. Write a C Program to simulate the working of a circular queue of integers using an array. Provide the following operations:

- a. Insert
- b. Delete
- c. Display

9. Write a C Program using dynamic variables and pointers, to construct a singly linked list consisting of the following information in each node: student id (integer), student name (character string) and semester (integer). The operations to be supported are:

- a. The insertion operation
  - i. At the front of a list
  - ii. At the back of the list
  - iii. At any position in the list

b. Deleting a node based on student id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.

c. Searching a node based on student id and update the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.

d. Displaying all the nodes in the list.

(Note: Only one set of operations among a, b and c with d may be asked in the examination)

10. Write a C Program using dynamic variables and pointers to construct a stack of integers using singly linked list and to perform the following operations:

- a. Push
- b. Pop
- c. Display

The program should print appropriate messages for stack overflow and stack empty.

11. Write a C program using dynamic variables and pointers to construct a queue of integers using singly linked list and to perform the following operations:

- a. Insert
- b. Delete
- c. Display

The program should print appropriate messages for queue full and queue empty.

12. Write a C Program to support the following operations on a doubly linked list where each node consists of integers:

- a. Create a doubly linked list by adding each node at the front.
- b. Insert a new node to the left of the node whose key value is read as an input
- c. Delete the node of a given data, if it is found, otherwise display appropriate message.
- d. Display the contents of the list.

(Note: Only either (a,b and d) or (a, c and d) may be asked in the examination)

13. Write a C Program

- a. To construct a binary search tree of integers.
- b. To traverse the tree using all the methods i.e., inorder, preorder and postorder.
- c. To display the elements in the tree.

14. Write recursive C Programs for

- a. Searching an element on a given list of integers using the Binary Search method.
- b. Solving the Towers of Hanoi problem.

Note: In the examination *each* student picks one question from a lot of *all* 14 questions.

**Electronic Circuits & Logic Design Laboratory**  
(Common to CSE & ISE)

Sub Code	: 06CSEL/06ISEL38	IA Marks	: 25
Hrs / Week	: 03	Exam Hours	: 03
Total Hrs	: 42	Exam Marks	: 50

**Part A**

1. a. To study the working of positive clipper, double-ended clipper and positive clamper using diodes.  
b. To build and simulate the above circuits using a simulation package
2. a. To determine the frequency response, input impedance, output impedance, and bandwidth of a CE amplifier.  
b. To build the CE amplifier circuit using a simulation package and determine the voltage gain for two different

values of supply voltage and for two different values of emitter resistance.

3. a. To determine the drain characteristics and transconductance characteristics of an enhancement-mode MOSFET.  
b. To implement a CMOS inverter using a simulation package and verify its truth table.
4. a. To design and implement a Schmitt trigger using Op-Amp for given UTP and LTP values.  
b. To implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values.
5. a. To design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency.  
b. To implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and observe the change in frequency when all resistor values are doubled.
6. To design and implement an astable multivibrator circuit using 555 timer for a given frequency and duty cycle.
7. To implement a +5V regulated power supply using full-wave rectifier and 7805 IC regulator in simulation package. Find the output ripple for different values of load current.

### Part B

1. a. Given any 4-variable logic expression, simplify using Entered Variable map and realize the simplified logic expression using 8:1 multiplexer IC.  
b. Write the Verilog/VHDL code for an 8:1 multiplexer. Simulate and verify its working.
2. a. Realize a full adder using 3-to-8 decoder IC and 4 input NAND gates.  
b. Write the Verilog/VHDL code for a full adder. Simulate and verify its working.
3. a. Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table.  
b. Write the Verilog/VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.
4. a. Design and implement a mod-n ( $n < 8$ ) synchronous up counter using J-K Flip-Flop ICs.  
b. Write the Verilog/VHDL code for mod-8 up counter. Simulate and verify its working.

5. a. Design and implement a ring counter using 4-bit shift register.
- b. Write the Verilog/VHDL code for switched tail counter. Simulate and verify its working.
6. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ( $n \leq 9$ ).
7. Design a 4-bit R-2R ladder D/A converter using Op-Amp. Determine its accuracy and resolution.

**Note** 1. Any simulation package like MultiSim/Pspice etc. may be used.

2. In the practical examination all students in batch will pick up one question from the diminishing lot that initially contained all the 14 questions.

**Graph Theory and Combinatorics**  
(Common to CSE & ISE)

<b>Sub Code</b>	: CSE/ISE 42	<b>IA Marks</b>	: 25
<b>Hrs / Week</b>	: 04	<b>Exam Hours</b>	: 03
<b>Total Hrs</b>	: 52	<b>Exam Marks</b>	: 100

**Part A**

Unit I

1. Introduction to Graph Theory: Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits

**7 Hours**

Unit II

2. Introduction to Graph Theory *contd.*: Planar Graphs, Hamilton Paths and Cycles, Graph Colouring, and Chromatic Polynomials

**6 Hours**

Unit III

3. Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes

**6 Hours**

Unit IV

4. Optimization and Matching: Dijkstra's Shortest Path Algorithm, Minimal Spanning Trees – The algorithms of Kruskal and Prim,

Transport Networks – Max-flow, Min-cut Theorem, Matching Theory

**7 Hours**

**Part B**

Unit V

5. Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition, The Catalan Numbers

**6 Hours**

Unit VI

6. The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials

**6 Hours**

Unit VII

7. Generating Functions: Introductory Examples, Definition and Examples – Computational Techniques, Partitions of Integers, The Exponential Generating Function, The Summation Operator

**7 Hours**

Unit VIII

8. Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients, The Non-homogeneous Recurrence Relation, The Method of Generating Functions

**7 Hours**

**Text Book**

1. Discrete and Combinatorial Mathematics, Ralph P. Grimaldi, 5<sup>th</sup> Edition, PHI/Pearson Education, 2004  
(Chapter 11, Chapter 12.1 to 12.4, Chapter 13, Chapter 1, Chapter 8.1 to 8.4, Chapter 9 Chapter 10.1 to 10.4)

**Reference Books**

1. Graph Theory and Combinatorics, Dr. D.S. Chandrasekharaiah, Prism, 2005
2. Introduction to Graph Theory, Chartrand Zhang, TMH, 2006
3. Introductory Combinatorics, Richard A. Brualdi, 4<sup>th</sup> Edition, Pearson Prentice Hall, 2004
4. Graph Theory Modeling, Applications, and Algorithms, Geir Agranarsson & Raymond Geenlaw, Pearson Prentice Hall, 2007

## Note

The Question paper consists of two parts A and B containing 4 questions each. The student is required to answer any 5 questions selecting at least two questions from each part.

### Analysis and Design of Algorithms (Common to CSE & ISE)

Sub Code	: CSE/ISE 43	IA Marks	: 25
Hrs / Week	: 04	Exam Hours	: 03
Total Hrs	: 52	Exam Marks	: 100

#### Unit I

1. Introduction: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental Data Structures

**6 Hours**

#### Unit II

2. Fundamentals of the Analysis of Algorithm Efficiency: Analysis Framework,
3. Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Nonrecursive and Recursive Algorithms, Example – Fibonacci Numbers

**6 Hours**

#### Unit III

4. Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search
5. Divide and Conquer: Mergesort, Quicksort, Binary Search

**6 Hours**

#### Unit IV

6. Divide and Conquer *contd.*: Binary tree traversals and related properties, Multiplication of large integers and Strassen's Matrix Multiplication.
7. Decrease and Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Algorithms for Generating Combinatorial Objects

**7 Hours**

#### Unit V

8. Transform and Conquer: Presorting, Balanced Search Trees, Heaps and Heapsort, Problem Reduction
9. Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching

**7 Hours**

**Unit VI**

10. Space and Time Tradeoff *contd.*: Hashing
11. Dynamic Programming: Computing a Binomial Coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem and Memory Functions

**6 Hours**

**Unit VII**

12. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees
13. Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees

**7 Hours**

**Unit VIII**

14. Limitations of Algorithm Power *contd.*: P, NP and NP-Complete Problems
15. Coping with the Limitations of Algorithm Power: Backtracking, Branch-and-Bound, Approximation Algorithms for NP-Hard Problems

**7 Hours**

**Text Book**

1. Introduction to The Design & Analysis of Algorithms, Anany Levitin, 2<sup>nd</sup> Edition, Pearson Education, 2007  
(Chapter 1, 2.1 to 2.5, 3.1; 3.2, 3.4, 4.1 to 4.5, 5.1 to 5.4, 6.1, 6.3, 6.4, 6.6, 7.1 to 7.3, 8.1, 8.2, 8.4, 9, 11.1, 11.2, 11.3, 12.1, 12.2, 12.3)

**Reference Books**

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 2<sup>nd</sup> Edition, PHI, 2006
2. Computer Algorithms by Horowitz E., Sahni S., Rajasekaran S., Galgotia Publications, 2001
3. Introduction to the Design and Analysis of Algorithms A Strategic Approach, R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T. Tsai, TMH, 2005

**Object Oriented Programming with C++**  
(Common to CSE & ISE)

**Sub Code : CSE/ISE 44**  
**Hrs / Week : 04**

**IA Marks : 25**  
**Exam Hours : 03**

**Total Hrs : 52**

**Exam Marks : 100**

**Unit I**

1. Introduction to C++: A Review of Structures, Procedure-Oriented Programming Systems, Object-Oriented Programming Systems, Comparison of C++ with C, Console Input/Output in C++, Variables in C++, Reference Variables in C++, Function Prototyping, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions

**4 Hours**

2. Class and Objects: Introduction to Classes and Objects

**2 Hours**

**Unit II**

3. Class and Objects *contd.*: Member Functions and Member Data, Objects and Functions, Objects and Arrays, Namespaces, Nested Classes

**6 Hours**

**Unit III**

4. Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation, The `set_new_handler()` function
5. Constructors and Destructors: Constructors, Destructors, The Philosophy of OOPS

**7 Hours**

**Unit IV**

6. Inheritance: Introduction to Inheritance, Base Class and Derived Class Pointers, Function Overriding, Base Class Initialization, The Protected Access Specifier, Deriving by Different Access Specifiers, Different Kinds of Inheritance, Order of Invocation of Constructors and Destructors

**6 Hours**

**Unit V**

7. Virtual Functions and Dynamic Polymorphism: The Need for Virtual Functions, Virtual Functions, The Mechanism of Virtual Functions, Pure Virtual Functions, Virtual Destructors and Virtual Constructors
8. Stream Handling: Streams, The Class Hierarchy of Handling Streams, Text and Binary Input/Output, Text Versus Binary Files, Text Input/Output, Binary Input/Output

**6 Hours**

**Unit VI**

9. Stream Handling *contd.*: Opening and Closing Files, Files as Objects of the `fstream` Class, File Pointer, Random Access to Files, Object Input/Output through Member Functions, Error Handling, Manipulators
10. Operator Overloading: Operator Overloading, Overloading the Various Operators – Overloading the Increment and the Decrement Operators (Prefix and Postfix), Overloading the Unary Minus and the Unary Plus Operator, Overloading the Arithmetic Operators

**7 Hours**

#### Unit VII

11. Operator Overloading *contd.*: Overloading the Relational Operators, Overloading the Assignment Operator, Overloading the Insertion and Extraction Operators, Overloading the `new` and the `delete` Operators, Overloading the Subscript Operator, Overloading the Pointer-to-member (`->`) Operator (Smart Pointer)

**7 Hours**

#### Unit VIII

12. Type Conversion, New Style Casts, and RTTI
13. Templates: Introduction, Function Templates, Class Templates, The Standard Template Library (STL)
14. Exception Handling: Introduction, C-Style Handling of Error-generating Codes, C++ Style Solution – the `try/throw/catch` Construct, Limitation of Exception Handling

**7 Hours**

#### Text Book

1. Object-Oriented Programming with C++, Sourav Sahay, Oxford University Press, 2006  
(Chapters 1 to 10)

#### Reference Books

1. C++ Primer, Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, 4<sup>th</sup> Edition, Addison Wesley, 2005
2. The Complete Reference C++, Herbert Schildt, 4<sup>th</sup> Edition, TMH, 2005

### Microprocessors

(Common to CSE & ISE)

<b>Sub Code</b>	<b>: CSE/ISE 45</b>	<b>IA Marks</b>	<b>: 25</b>
<b>Hrs / Week</b>	<b>: 04</b>	<b>Exam Hours</b>	<b>: 03</b>
<b>Total Hrs</b>	<b>: 52</b>	<b>Exam Marks</b>	<b>: 100</b>

Unit I

1. Overview of Microcomputer Structure and Operation, Microprocessor Evolution and Types, 8086 Internal Architecture, Introduction to Programming the 8086

**6 Hours**

Unit II

2. 8086 Family Assembler Language Programming – Instruction Templates, MOV Instruction Coding Format and Examples, MOV Instruction Coding Examples, Writing Programs for use with an Assembler, Assembly Language Program Development Tools

**6 Hours**

Unit III

3. Implementing Standard Program Structures In 8086 Assembly Language: Simple Sequence Programs, Jumps, Flags, and Conditional Jumps, If-Then, If-Then-Else, and Multiple If-Then-Else Programs, While-Do Programs, Repeat-Until Programs, Instruction Timing and Delay Loops

**7 Hours**

Unit IV

4. Strings, Procedures, and Macros: The 8086 String Instructions, Writing and Using Procedures, Writing and Using Assembler Macros

**7 Hours**

Unit V

5. 8086 Instruction Description and Assembler Directives

**6 Hours**

Unit VI

6. 8086 System Connections Timing: A Basic 8086 Microcomputer System, Addressing Memory and Ports in Microcomputer Systems, 8086 and 8088 Addressing and Address Decoding, How the 8088 Microprocessor Accesses Memory and Ports, 8086 Timing Parameters

**7 Hours**

Unit VII

7. 8086 Interrupts and Interrupt Applications: 8086 Interrupts and Interrupt Responses, Hardware Interrupt Applications, 8259A Priority Interrupt Controller, Software Interrupt Applications

**7 Hours**

Unit VIII

8. Digital Interfacing: Programmable Parallel Ports and Handshake Input/Output, Methods of Data Transfer, Implementing Handshake Data Transfer, 8255A Internal Block Diagram and System

Connections, 8255A Operational Modes and Initialization, Constructing and Sending 8255A Control Words

**6 Hours**

**Text Book**

1. Microprocessors and Interfacing, Douglas V. Hall, Revised 2<sup>nd</sup> Edition, TMH

**Reference Books**

1. Advanced Microprocessors & IBM-PC assembly Language Programming, K. Udaya Kumar & B.S. Umashankar, TMH 2003
2. The Intel Microprocessors, 7<sup>th</sup> Edition, Barry B. Brey, Pearson/PHI 2006  
(Section 4-7, Section 8-1)
3. The Intel Microprocessor Family: Hardware and Software Principles and Applications, James L. Antonakos, Thomson, 2007

**Computer Organization**

*(Common to CSE & ISE)*

<b>Sub Code</b>	<b>: CSE/ISE 46</b>	<b>IA Marks</b>	<b>: 25</b>
<b>Hrs / Week</b>	<b>: 04</b>	<b>Exam Hours</b>	<b>: 03</b>
<b>Total Hrs</b>	<b>: 52</b>	<b>Exam Marks</b>	<b>: 100</b>

**Unit I**

1. Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement, Historical Perspective
2. Machine Instructions and Programs: Numbers, Arithmetic Operations and Characters, Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing,

**6 Hours**

**Unit II**

3. Machine Instructions and Programs *contd.:* Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions

**7 Hours**

Unit III

4. Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses

**6 Hours**

Unit IV

5. Input/Output Organization *contd.*: Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB

**6 Hours**

Unit V

6. Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations

**6 Hours**

Unit VI

7. Memory System *contd.*: Virtual Memories, Secondary Storage
8. Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders

**7 Hours**

Unit VII

9. Arithmetic *contd.*: Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division, Floating-point Numbers and Operations

**7 Hours**

Unit VIII

10. Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Microprogrammed Control

**7 Hours**

Text Book

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5<sup>th</sup> Edition, TMH, 2002  
(Chapter 1.1 to 1.4, 1.6.1, 1.6.2, 1.6.4, 1.6.7, 1.8, Chapter 2.1 to 2.10, 2.12, Chapter 4.1, 4.2.1 to 4.2.5, 4.4 to 4.7, Chapter 5.1 to 5.4, 5.5.1, 5.5.2, 5.6, 5.7, 5.9, Chapter 6, 7)

**Reference Books**

1. Computer Organization & Architecture, William Stallings, 7<sup>th</sup> Edition, PHI, 2006

2. Computer Systems Design and Architecture, Vincent P. Heuring & Harry F. Jordan, 2<sup>nd</sup> Edition, Pearson Education, 2004

**Object Oriented Programming with C++ Laboratory**  
(Common to CSE & ISE)

Sub Code	: CSEL/ISEL 47	IA Marks	: 25
Hrs / Week	: 03	Exam Hours	: 03
Total Hrs	: 42	Exam Marks	: 50

1. Given that an EMPLOYEE class contains following members:  
Data members : Employee\_Number, Employee\_Name, Basic, DA, IT, Net\_Salary  
Member functions: to read the data, to calculate Net\_Salary and to print data members. Write a C++ program to read the data of N employees and compute Net\_Salary of each employee.  
(Dearness Allowance (DA) = 52% of Basic and Income Tax (IT) = 30% of the gross salary. Net\_Salary = Basic + DA - IT)
2. Define a STUDENT class with USN, Name, and Marks in 3 tests of a subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of two better marks for each student. Print the USN, Name, and the average marks of all the students.
3. Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a COMPLEX number.
- ADD (a, s2) – where a is an integer (real part) and s2 is a complex number.
  - ADD (s1, s2) – where s1 and s2 are complex numbers.
4. Write a C++ program to create a class called LIST (linked list) with member functions to insert an element at the front of the list as well as to delete an element from the front of the list. Demonstrate all the functions after creating a list object.
5. Write a C++ program to create a template function for Quick sort and demonstrate sorting of integers and doubles.

6. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators + and -.

i.  $s1 = s1 + \text{element}$ ; where  $s1$  is an object of the class STACK and element is an integer to be pushed on to top of the stack.

ii.  $s1 = s1 -$  ; where  $s1$  is an object of the class STACK and  $-$  operator pops the element.

Handle the STACK Empty and STACK Full conditions. Also display the contents of the stack after each operation, by overloading the operator <<.

7. Write a C++ program to create a class called DATE. Accept two valid dates in the form dd/mm/yy. Implement the following operations by overloading the operators + and -. After every operation display the results by overloading the operator <<.

i.  $\text{no\_of\_days} = d1 - d2$  where  $d1$  and  $d2$  are DATE objects,  $d1 \geq d2$  and no\_of\_days is an integer.

ii.  $d2 = d1 + \text{no\_of\_days}$ ; where  $d1$  is a DATE object and no\_of\_days is an integer.

8. Write a C++ program to create a class called MATRIX using a two-dimensional array of integers. Implement the following operations by overloading the operator == which checks the compatibility of two matrices  $m1$  and  $m2$  to be added and subtracted. Perform the addition and subtraction by overloading the operators + and - respectively. Display the results (sum matrix  $m3$  and difference matrix  $m4$ ) by overloading the operator <<.

```
if(m1 == m2)
{
    m3 = m1 + m2;
    m4 = m1 - m2;
}
else
display error
```

9. Write a C++ program to create a class called OCTAL, which has the characteristics of an octal number. Implement the

following operations by writing an appropriate constructor and an overloaded operator +.

i. OCTAL  $h = x$  ; where  $x$  is an integer

ii.  $\text{int } y = h + k$  ; where  $h$  is an OCTAL object and  $k$  is an integer.

Display the OCTAL result by overloading the operator <<. Also display the values of  $h$  and  $y$ .

10. Write a C++ program to create a class called QUEUE with member functions to add an element and to delete an element from the queue. Using these member functions, implement a queue of integers and doubles. Demonstrate the operations by displaying the contents of the queue after every operation.

11. Write a C++ program to create a class called DLIST (Doubly Linked List) with member functions to insert a node at a specified position and delete a node from a specified position of the list. Demonstrate the operation by displaying the contents of the list after every operation.

12. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the semester wise average age for all UG and PG students separately.

13. Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.

i. STRING  $s1 = \text{"VTU"}$

ii. STRING  $s2 = \text{"BELGAUM"}$

iii. STRING  $s3 = s1 + s2$  ; (Use copy constructor).

14. Write a C++ program to create a class called BIN\_TREE ( Binary tree) with member functions to perform inorder, preorder and postorder traversals. Create a BIN\_TREE object and demonstrate the traversals.

15. Write a C++ program to create a class called EXPRESSION. Using appropriate member functions convert a given valid Infix expression into Postfix form. Display the Infix and Postfix expressions.

Note: In the examination *each* student picks one question from a lot of *all* 15 questions.

**Microprocessor Laboratory**  
(Common to CSE and ISE)

Sub Code	: CSEL/ISEL 48	IA Marks	: 25
Hrs / Week	: 03	Exam Hours	: 03
Total Hrs	: 42	Exam Marks	: 50

**Note:**

- Develop and execute the following programs using an 8086 Assembly Language. All the programs to be executed using an assembler like MASM/TASM etc.
- Program should have suitable comments.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.

1. a) Search a key element in a list of 'n' 16-bit numbers using the Binary search algorithm.

b) Read the status of eight input bits from the Logic Controller Interface and display 'FF' if it is even parity bits otherwise display '00'. Also display number of 1's in the input data.

2. a) Write ALP macros:

i. To read a character from the keyboard in the module (1) (in a different file)

ii. To display a character in module(2) (from different file)

iii. Use the above two modules to read a string of characters from the keyboard terminated by the carriage return and print the string on the display in the next line.

b) Perform the following functions using the Logic Controller Interface.

i. BCD up-down Counter ii. Ring Counter

3. a) Sort a given set of 'n' numbers in ascending and descending orders using the Bubble Sort algorithm.

- b) Read the status of two 8-bit inputs (X & Y ) from the Logic Controller Interface and display  $X*Y$ .
4. a) Read an alphanumeric character and display its equivalent ASCII code at the center of the screen.  
b) Display messages FIRE and HELP alternately with flickering effects on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor it is necessary for the student to compute these values).
5. a) Reverse a given string and check whether it is a palindrome or not.  
b) Assume any suitable message of 12 characters length and display it in the rolling fashion on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages. (Examiner does not specify these delay values nor it is necessary for the student to compute these values).
6. a) Read two strings, store them in locations STR1 and STR2. Check whether they are equal or not and display appropriated messages. Also display the length of the stored strings.  
b) Convert a 16-bit binary value (assumed to be an unsigned integer) to BCD and display it from left to right and right to left for specified number of times on a 7-segment display interface.
7. a) Read your name from the keyboard and display it at a specified location on the screen in front of the message **What is your name?** You must clear the entire screen before display.  
b) Drive a Stepper Motor interface to rotate the motor in clockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).
8. a) Compute the factorial of a positive integer 'n' using recursive procedure.

- b) Drive a stepper motor interface to rotate the motor in anti-clockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps (Any arbitrary value for the delay may be assumed by the student).
9. a) Compute  $nCr$  using recursive procedure. Assume that 'n' and 'r' are non-negative integers.  
b) Drive a stepper motor interface to rotate the motor by N steps left direction and N steps right direction (N is specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).
10. a) Find out whether a given sub-string is present or not in a main string of characters.  
b) Scan a 8 x 3 keypad for key closure and to store the code of the key pressed in a memory location or display on screen. Also display row and column numbers of the key pressed.
11. a) Generate the first 'n' Fibonacci numbers.  
b) Scan a 8 x 3 keypad for key closure and simulate ADD and SUBTRACT operations in a calculator.
12. a) Read the current time from the system and display it in the standard format on the screen.  
b) Generate the sine wave using DAC interface (The output of the DAC is to be displayed on the CRO).
13. a) Program to simulate a Decimal Up-counter to display 00-99.  
b) Generate a Half Rectified Sine wave form using the DAC interface. (The output of the DAC is to be displayed on the CRO).
14. a) Read a pair of input co-ordinates in BCD and move the cursor to the specified location on the screen.  
b) Generate a Fully Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).
15. a) Program to create a file (input file) and to delete an existing file.  
b) Drive an elevator interface in the following way:

- i. Initially the elevator should be in the ground floor, with all requests in OFF state.
- ii. When a request is made from a floor, the elevator should move to that floor, wait there for a couple of seconds, and then come down to ground floor and stop. If some requests occur during going up or coming down they should be ignored.

Note: In the examination *each* student picks one question from a lot of *all* 15 questions.

\*\*\*\*\*

COMPUTER SCIENCE & ENGINEERING