



माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA

Deemed University

(Declared under Distinct Category by Ministry of Education, Government of India)

NAAC ACCREDITED WITH A++ GRADE



Department of Engineering Mathematics and Computing

Department of Engineering Mathematics & Computing Syllabus I Sem.



Department of Engineering Mathematics and Computing

B. Tech. (First Semester) Introduction to Computing (25241101)

Course Objectives:

- Explain the needs of hardware and software required for a computation task.
- State typical provisions of cyber law that govern the proper usage of Internet and computing resources.
- Explain the working of important application software and their use to perform any engineering activity.
- Understand the Importance of Operating system & its requirement.

UNIT 1: Computer: Definition, Classification, Organization i.e. CPU, register, Bus architecture, Instruction set, Memory & Storage Systems, I/O Devices, and System & Application Software, Operating System: Definition, Function, Types, Management of File, Process & Memory.

UNIT 2: Computer Networking: Introduction, Introduction to Internet, World Wide Web, E-commerce Computer Security Basics: Introduction to viruses, worms, malware, Trojans, Spyware and Anti-Spyware Software, Different types of attacks like Money Laundering, Information Theft, Email spoofing, Denial of Service (DoS), Cyber Stalking, Logic bombs, Hacking Spamming, Cyber Defamation, phishing Security measures Firewall, Computer Ethics & Good Practices

UNIT 3: Data base Management System: Introduction, File oriented approach and Database approach, Data Models, Architecture of Database System, Data independence, Data dictionary, DBA, Primary Key, Data definition language and Manipulation Languages.

UNIT 4: Cloud computing: definition, cloud infrastructure, cloud segments or service delivery models (IaaS, PaaS and SaaS), cloud deployment models/ types of cloud (public, private, community and hybrid clouds), Pros and Cons of cloud computing

UNIT 5: Career opportunities Entrepreneurship, Start up: Scope, Funding Opportunities, Other career opportunities; Case Studies Success stories of Google, Facebook, Flipkart, Amazon, Ola, Uber etc. in socio-economic progress

| CO's | Description of CO's |
|------|--|
| CO1 | Defining core components of computing and linkage between them, |
| CO2 | Summarizing role of operating system |
| CO3 | Discuss the role of computing in real world applications |
| CO4 | Explain Networking aspect of computer engineering and communication, |
| CO5 | Acquire basic knowledge of database system |
| CO6 | Explain the Fundamentals of Entrepreneurship and Start up |

RECOMMENDED BOOKS:

1. Computer Science: An Overview, J. Glenn Brookshear, and Dennis Brylow, Pearson.
2. Fundamentals of Computers, V. Rajaraman, and Neeharika Adabala, PHI.
3. Fundamentals of Computers: E Balagurusamy, TMH
4. Introduction of Computers: Peter Norton, TMH



Department of Engineering Mathematics and Computing

B. Tech. (First Semester) Computer Programming (25241102)

Course Objective:

- Develop ability to write a computer program to solve specified problems.
- Develop skills in algorithmic problem-solving, expressed in a programming language like C++.
- Learn what computer science is about, especially hardware, data representations, algorithms, encodings, form of programming.
- Understand fundamentals of programming such as variables, conditional and iterative statement, function and its execution etc.

UNIT 1: Introduction to Programming, Machine Level Languages, Assembly Level Languages, High Level Languages, Program Execution and Translation Process, Problem solving using Algorithms and Flowcharts. Introduction to C++ Programming: Data Types, Constants, Keywords, Operators & Expressions, Precedence of operators and input/output functions.

UNIT 2: Control Statements and Decision Making: The if statement, The if-else statement, Nesting of if statements, The conditional expression, The switch statement, The while loop, The do...while loop, The for loop, The nesting of for loops, The break and continue statement.

UNIT 3: Arrays, Strings & Pointers: One dimensional Arrays, Passing Arrays to Functions, Multidimensional Arrays, Strings, Basics of Pointers & Addresses, Pointer to Pointer, Pointer to Array, Array of Pointers, Types of pointers, Pointer to Strings.

UNIT 4: Functions & Structures: Function Basics, Function Prototypes, Passing Parameter by value and by reference, Passing string to function, Passing array to function, Function returning address, Recursion, Structures & Union, Pointer to Structure, Self-Referential Structures, Dynamic memory allocation by call of function, Storage Classes.

UNIT 5: File Handling: Defining and Opening a file, Closing Files, Input/output Operations on Files, Predefined Streams, Error Handling during I/O Operations, Command Line Arguments. Preprocessor Directives. Formatted I/O.

Course Outcomes

| CO's | Description of CO's |
|------|--|
| CO1 | Describe the basic principles of imperative and structural programming. |
| CO2 | Develop a pseudo-code and flowchart for a given problem. |
| CO3 | Analyze the problems and choose suitable programming techniques to develop solutions |
| CO4 | To design, implement, debug and test programs |
| CO5 | Design computer programs to solve real world problems |

Books:

1. E. Balagurusamy, "Programming in ANSI C++", Seventh Edition, Tata McGraw Hill, 2017.
2. Reema Thareja, "Programming in C++", Second Edition, Oxford publication, 2016
3. W. Kernighan and Dennis M. Ritchie, "The C/C++ Programming Language", Pearson, 2015
4. Matthias Felleisen, Robert Bruce Findler, Mathew Flatt, Shriram Krishnamurthi, "How to Design Programs: An Introduction to Programming and Computing", Second Edition, MIT Press, 2018.
5. E. Balagurusamy, "Object Oriented Programming with C++", Tata McGraw Hill, 2009
6. B.S. Gottfried, "Programming with C++", 3rd edition, Tata McGraw Hill, 2018

Web-resources:

1. https://www.tutorialspoint.com/cprogramming/c_useful_resources.htm
2. <https://github.com/kspalaiologos/C-Learning-Resources>
3. <https://www.geeksforgeeks.org/c-programming-language>
4. <https://leetcode.com>



Department of Engineering Mathematics and Computing

B. Tech. (First Semester) Advanced Calculus (25241103)

Objective of Course

- To understand the basic concepts of differential calculus
- To explore the applications of derivatives
- To familiarize the integral calculus
- To describe multiple integral
- To understand the concepts of Convergence and divergence

UNIT 1: Maclaurin's and Taylor's theorem, Partial differentiation, Euler's theorem, Deduction of Euler's Theorem, Jacobian and its properties, Maxima and Minima of one and two variables,

UNIT 2: Rolle's Theorem, First mean value theorem, Second mean value theorem, Curvature, radius of curvature, Asymptotes of Cartesian and Polar forms.

UNIT 3: Definite integral as limit of a sum, application in summation of series, Improper integral, Beta and Gamma function and its properties, some transformation of Beta function, some transformations of Gamma function, relation between Beta and Gamma function, Legendre's duplication formula

UNIT 4: Multiple integral and their applications, Double and Triple integral, Change of order of integration, Length of the curves, Volumes and Surfaces of solids of revolution.

UNIT 5: Concept of convergence and divergence, Basic test of convergence for sequence and series, P-Series test, Ratio test, Comparison test, Integral test, Cauchy's root test, Test of convergence and divergence of improper integral.

Course Outcomes

After completing this course, the students will be able to:

| CO's | Description of CO's |
|------|---|
| CO1 | Determine the solution of function by using one and two variables. |
| CO2 | Interpret the solution of derivatives concepts using different techniques |
| CO3 | Evaluate the Gamma and Beta Function |
| CO4 | Obtain the volume and area of surface by using multiple integrals |
| CO5 | Identify the convergence and divergence of sequence and series |

Recommended Books:

1. E. Kreyszig: Advance Engineering Mathematics, John Wiley & Sons, 10th Edition (2011).
2. R. K. Jain, S. R. K. Iyengar: Advance Engineering Mathematics, Narosa Publishing House Pvt.Ltd, 5th Edition (2016).
3. F. B. Hildebrand: Advanced Calculus for application, Englewood Cliffs, N. J. Prentice- Hall, 2nd Edition (1980).
4. B. S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 43rd Edition (2015).
5. B.V. Ramanna: Higher Engineering Mathematics, McGraw Hill Education, 1st Edition (2017).



Department of Engineering Mathematics and Computing

B. Tech. (First Semester) Digital Logic Design (25241104)

Course Objective

- To familiarize with number representation and conversion between various representations in digital electronic circuits.
- To expose the logical operations using combinational logic circuits, sequential logic circuits and the characteristics of memory and their classification.

Unit 1: Introduction to Digital Electronics, Needs and Significance, Different Number System: Binary, Decimal, Octal and Hexadecimal Numbers, Conversions, Complement's, Signed Binary Numbers, Binary Arithmetic's, Binary Codes: BCD, ASCII Codes.

Unit 2: Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Boolean Relations, Digital Logic Gates, De Morgan's Theorem, Karnaugh Maps and simplifications.

Unit 3: Combinational Circuits, Half Adder, Full Adder, Binary Adder-Subtractor, Binary Multiplier, Comparator, Decoders, Encoders, Multiplexers.

Unit 4: Sequential Circuits, Latches, Flip-Flops: RS Latches, Level Clocking, D Latches, Edge-Triggered D Flip-flop, Edge-Triggered JK Flip-flop, JK Master-Slave Flip-flop; Registers, Shift Registers, Counters, Ripple Counters, Synchronous Counters.

Unit 5: Introduction to Memory, Classification of Memories, Memory Decoding, Programmable Devices: Programmable Logic Array (PLA), Programmable Array Logic (PAL).

Course Outcomes

After completion of the course students would be able to:

| CO's | Description of CO's |
|------|--|
| CO1 | Explain different number systems and conversion among them and codes |
| CO2 | Simplify the logic expressions using Boolean laws, and map methods and design them by using logic gates. |
| CO3 | Develop the understanding of combinational circuits and design them. |
| CO4 | Analyze different types of flip-flops and design a sequential logic circuit |
| CO5 | Compare various memories used in computers |

Recommended Books

- Fundamentals of Digital Logic Design by Charles H. Roth, Jr. Cengage, 7th Edition.
- Digital Design, Morris Mano M. and Michael D. Ciletti, Pearson Education, 6th Edition.
- Digital Electronics: Principles, Devices and Applications, Anil K. Maini, Wiley.



Department of Engineering Mathematics and Computing

B. Tech. (First Semester) Differential Equations (25241105)

Course Objective

- To understand the concept of Ordinary differential equations
- To familiarize the solution techniques of ODE of higher order with constant and variable coefficients
- To describe how to form partial differential equation
- To explore various type of partial differential equation and its application

UNIT 1: Ordinary differential equations of first and higher order, Differential equations in which variables are separable, Homogeneous differential equations, Differential equation reducible to homogeneous form (Non-homogeneous differential of first degree), linear differential equation (Leibnitz's linear differential equation), Bernoulli's equation of differential equation reducible to linear form, exact differential equations.

UNIT-2: Linear higher order differential equation with constant coefficients, Homogeneous linear equations or Cauchy's Euler's equations, Solution of simultaneous differential equations.

UNIT 3: Second order differential equations with variable coefficients, Methods: one integral is known, Removable of first derivative, changing of independent variable and variation of parameters, Solution of Differential equation by Series Solution method.

UNIT 4: Introduction of partial differential equation, Formulation of partial differential equation, Linear Partial differential equations of first order and solution techniques Lagrange's method, and Non-Linear Partial differential equations of first order and standard form I, II, III & IV and Charpit's method.

UNIT-5: Partial differential equations of higher order with constant coefficients, Homogeneous and Non-Homogeneous Linear Partial differential equations, Classification of Partial differential equations, Application of Partial differential equations to solve wave equation and heat equation (one-dimensional) by Separation of variables method.

Course Outcomes

After completing this course, the students will be able to:

| CO's | Description of CO's |
|------|---|
| CO1 | Determine the analytic solution of ordinary differential equations |
| CO2 | Interpret the solution of ordinary differential equations with constant and variable coefficients |
| CO3 | Acquire the knowledge of second and higher order differential equation |
| CO4 | Formulate the Partial differential equations |
| CO5 | Evaluate the Partial differential equations of higher order with its application |

Recommended Books:

1. E. Kreyszig: Advance Engineering Mathematics, John Wiley & Sons, 10th Edition (2011).
2. R. K. Jain, S. R. K. Iyengar: Advance Engineering Mathematics, Narosa Publishing House Pvt. Ltd., 5th Edition (2016).
3. B. S. Grewal: Higher Engineering Mathematics, Khanna Publisher, 43rd Edition (2015).
4. H. K. Dass: Advance Engineering Mathematics, S. Chand Publisher (2018).
5. B.V. Ramanna: Higher Engineering Mathematics, McGraw Hill Education, 1st Edition (2017).



Department of Engineering Mathematics and Computing

B. Tech. (First Semester)

Universal Human Values & Professional Ethics (25241111)

Course Objectives

- Sensitization of student towards self, family (relationship), society and nature.
- Understanding (or developing clarity) of nature, society and larger systems, on the basis of human relationships and resolved individuals.
- Strengthening of self-reflection.
- Development of commitment and courage to act.

Course Content:

UNIT 1: Course Introduction - Need, Basic Guidelines, Content and Process for Value Education:

- Self-Exploration—what is it? - Its content and process; ‘Natural Acceptance’ and Experiential Validation- as the process for self-exploration
- Continuous Happiness and Prosperity- A look at basic Human Aspirations
- Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority
- Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario

UNIT 2: Understanding Harmony in the Human Being:

- Understanding human being as a co-existence of the sentient ‘I’ and the material ‘Body’
- Understanding the needs of Self (‘I’) and ‘Body’ - happiness and physical facility
- Understanding the Body as an instrument of ‘I’ (I being the doer, seer and enjoyer)
- Understanding the characteristics and activities of ‘I’ and harmony in ‘I’
- Understanding the harmony of ‘I’ with the Body

UNIT 3: Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship:

- Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship
- Understanding the meaning of Trust; Difference between intention and competence
- Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship
- Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals
- Visualizing a universal harmonious order in society

UNIT 4: Understanding Harmony in the Nature and Existence - existence as Coexistence:

- Understanding the harmony in the Nature
- Interconnectedness and mutual fulfilment among the four orders of nature recyclability and self-regulation in nature
- Understanding Existence as Co-existence of mutually interacting units in all pervasive space
- Holistic perception of harmony at all levels of existence.

UNIT 5: Holistic Understanding of Harmony on Professional Ethics:

- Natural acceptance of human values
- Definitiveness of Ethical Human Conduct
- Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order
- Competence in professional ethics:
 - a. Ability to utilize the professional competence for augmenting universal human order
 - b. Ability to identify the scope and characteristics of people friendly and eco-friendly production systems,
 - c. Ability to identify and develop appropriate technologies and management patterns for above production systems.



Department of Engineering Mathematics and Computing

- Strategy for transition from the present state to Universal Human Order:
 - a. At the level of individual: as socially and ecologically responsible engineers, technologists and managers
 - b. At the level of society: as mutually enriching institutions and organizations

Gender Sensitisation:

- Introduction to Sex, Gender & Culture
- Introduction to Women Studies and Socialisation, including man-woman relationship, work distribution
- A brief review of Feminism, Patriarchy, Feminist Studies, Feminist Ideologies.
- Women and Law Constitutional Provisions and Fundamental rights related to Women.

Course Outcomes:

At the end of the course student will be able

| CO's | Description of CO's |
|------|---|
| CO1 | to become more aware of their surroundings, society, social problems and their sustainable solutions |
| CO2 | to become sensitive to their commitment towards what they believe in (humane values, humane relationships and humane society) |
| CO3 | to apply what they have learnt to their own self in different day-to-day settings in real life |
| CO4 | to sustain human relationships and human nature in mind |
| CO5 | to have better critical ability |
| CO6 | to negotiate living in harmony with self and others |

Text Book

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010

Reference Books

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. On Education - J Krishnamurthy
6. Siddhartha - Hermann Hesse
7. Old Path White Clouds - Thich Nhat Hanh
8. On Education - The Mother
9. Diaries of Anne Frank - Anne Frank
10. Life and Philosophy of Swami Vivekananda
11. Swami Vivekananda on Himself
12. Small is Beautiful - E. F Schumacher.
13. Slow is Beautiful - Cecile Andrews
14. Economy of Permanence - J C Kumarappa
15. Bharat Mein Angreji Raj - Pandit Sunderlal
16. Mahatma and the Rose
17. The Poet and the Charkha
18. Rediscovering India - by Dharampal
19. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
20. Swaraj by Arvind Kejriwal
21. India Wins Freedom - Maulana Abdul Kalam Azad
22. Ramakrishna ki jeevani - Romain Rolland (English)
23. Vivekananda - Romain Rolland (English)
24. Gandhi - Romain Rolland (English)
25. Autobiography of a Yogi – by Paramhansa Yogananda
26. Gandhi and Question of Science – Sahatsrabudhe



Department of Engineering Mathematics and Computing

B. Tech. (First Semester)

Computer Programming Lab (25241105)

Experiment List

1. Write a C++ "Hello, World!" Program
2. Write a C++ Program to Print an Integer (Entered by the User)
3. Write a C++ Program to Add Two Integers
4. Write a C++ Program to Multiply Two Floating-Point Numbers
5. Write a C++ Program to Find ASCII Value of a Character
6. Write a C++ Program to Compute Quotient and Remainder
7. Write a C++ Program to Find the Size of int, float, double and char
8. Write a C++ Program to Demonstrate the Working of Keyword long
9. Write a C++ Program to Swap Two Numbers
10. Write a C++ Program to Check Whether a Number is Even or Odd
11. Write a C++ Program to Check Whether a Character is a Vowel or Consonant
12. Write a C++ Program to Find the Largest Number Among Three Numbers
13. Write a C++ Program to Find the Roots of a Quadratic Equation
14. Write a C++ Program to Check Leap Year
15. Write a C++ Program to Check Whether a Number is Positive or Negative
16. Write a C++ Program to Check Whether a Character is an Alphabet or not
17. Write a C++ Program to Calculate the Sum of Natural Numbers
18. Write a C++ Program to Find Factorial of a Number
19. Write a C++ Program to Generate Multiplication Table
20. Write a C++ Program to Display Fibonacci Sequence
21. Write a C++ Program to Find GCD of two Numbers
22. Write a C++ Program to Find LCM of two Numbers
23. Write a C++ Program to Display Characters from A to Z Using Loop
24. Write a C++ Program to Count Number of Digits in an Integer
25. Write a C++ Program to Reverse a Number
26. Write a C++ Program to Calculate the Power of a Number
27. Write a C++ Program to Check Whether a Number is Palindrome or Not
28. Write a C++ Program to Check Whether a Number is Prime or Not
29. Write a C++ Program to Display Prime Numbers Between Two Intervals
30. Write a C++ Program to Check Armstrong Number
31. Write a C++ Program to Display Armstrong Number Between Two Intervals
32. Write a C++ Program to Display Factors of a Number
33. Write a C++ Program to Make a Simple Calculator Using switch...case
34. Write a C++ Program to Display Prime Numbers Between Intervals Using Function
35. Write a C++ Program to Check Prime or Armstrong Number Using User-defined Function
36. Write a C++ Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers
37. Write a C++ Program to Find the Sum of Natural Numbers using Recursion
38. Write a C++ Program to Find Factorial of a Number Using Recursion
39. Write a C++ Program to Find G. Write a C++.D Using Recursion



Department of Engineering Mathematics and Computing

40. Write a C++ Program to Convert Binary Number to Decimal and vice-versa
41. Write a C++ Program to Convert Octal Number to Decimal and vice-versa
42. Write a C++ Program to Convert Binary Number to Octal and vice-versa
43. Write a C++ Program to Reverse a Sentence Using Recursion
44. Write a C++ program to calculate the power using recursion
45. Write a C++ Program to Calculate Average Using Arrays
46. Write a C++ Program to Find Largest Element in an Array
47. Write a C++ Program to Calculate Standard Deviation
48. Write a C++ Program to Add Two Matrices Using Multi-dimensional Arrays
49. Write a C++ Program to Multiply Two Matrices Using Multi-dimensional Arrays
50. Write a C++ Program to Find Transpose of a Matrix
51. Write a C++ Program to Multiply two Matrices by Passing Matrix to a Function
52. Write a C++ Program to Access Array Elements Using Pointer
53. Write a C++ Program Swap Numbers in Cyclic Order Using Call by Reference
54. Write a C++ Program to Find Largest Number Using Dynamic Memory Allocation
55. Write a C++ Program to Find the Frequency of Characters in a String
56. Write a C++ Program to Count the Number of Vowels, Consonants and so on
57. Write a C++ Program to Remove all Characters in a String Except Alphabets
58. Write a C++ Program to Find the Length of a String
59. Write a C++ Program to Concatenate Two Strings
60. Write a C++ Program to Copy String Without Using strcpy()
61. Write a C++ Program to Sort Elements in Lexicographical Order (Dictionary Order)
62. Write a C++ Program to Store Information of a Student Using Structure
63. Write a C++ Program to Add Two Distances (in inch-feet system) using Structures
64. Write a C++ Program to Add Two Complex Numbers by Passing Structure to a Function
65. Write a C++ Program to Calculate Difference Between Two Time Periods
66. Write a C++ Program to Store Information of Students Using Structure
67. Write a C++ Program to Store Data in Structures Dynamically
68. Write a C++ Program to Write a Sentence to a File
69. Write a C++ Program to Read the First Line From a File
70. Write a C++ Program to Display its own Source Code as Output
71. Write a C++ Program to Print Pyramids and Patterns



Department of Engineering Mathematics and Computing

B. Tech. (First Semester)

Computing Lab for Data Analysis (25241107)

UNIT 1: Introduction to MS Excel - MS Excel Options – Ribbon - Sheets - Saving Excel File as PDF, CSV and Older versions - Using Excel Shortcuts - Copy, Cut, Paste, Hide, Unhide, and Link the Data in Rows, Columns and Sheet - Using Paste Special Options - Formatting Cells, Rows, Columns and Sheets - Protecting & Unprotecting Cells, Rows, Columns and Sheets with or without Password - Page Layout and Printer Properties.

UNIT 2: Functions: - Logical Functions - Date and Time Functions - Information Functions - Math and Trigonometry Functions - Statistical Functions - Text Functions - Charts:- Simple Bar Chart – Multiple Bar Chart – Subdivided Bar Chart – Pie Chart – Donut Chart - Line Chart – Histogram – Scatter Plot - Radar Chart – Bubble Chart – Bi-Axis chart – Plotting Density Function and Distribution Function. Vlookup, Hlookup, Index, Address, Match, Offset, Transpose - Conditional Formatting - Data Sorting and Filtering - Pivot Tables - Chart Templates – Adding Add-Ins in Excel - Solver – Goal Seek.

UNIT 3: Data handling: open SPSS data file – save – import from other data source – data entry – labelling for dummy numbers - recode in to same variable – recode in to different variable – transpose of data – insert variables and cases – merge variables and cases.

UNIT 4: Data handling: Split – select cases – compute total scores – table looks – Changing column – font style and sizes. Diagrammatic representation: Simple Bar diagram – Multiple bar diagram – Sub-divided Bar diagram, Percentage diagram - Pie Diagram – Frequency Table – Histogram – Scatter diagram – Box plot.

UNIT 5: Application to excel and Spss for Descriptive Statistics - Mean, Median, Mode, SD- Skewness- Kurtosis. Correlation – Karl Pearson's and Spearman's Rank Correlation , Regression analysis: Simple and Multiple Regression Analysis [Enter and stepwise methods]

List of Experiments

1. Lab 1 Introduction to Microsoft Excel
2. Lab 2 Frequency Distributions and Graphs
3. Lab 3 Data Description
4. Lab 4 Probability and Counting Rules
5. Lab 5 Discrete Probability Distributions
6. Lab 6 The Normal Distribution
7. Lab 7 Confidence Intervals
8. Lab 8 Hypothesis Testing
9. Lab 9 Testing the Difference Between Parameters from Two Populations
10. Lab 10 Correlation and Regression
11. Lab 11 Tests for Categorical Variables
12. Lab 12 One-Way Analysis of Variance (ANOVA)
13. Lab 13 Stem-and-Leaf Plots and Frequency Tables
14. Lab 14 Summary Statistics USING SPSS.
15. Lab 15 To calculate and interpret binomial and normal probabilities.
16. Lab 16 Testing a Mean.
17. Lab 17 Paired Samples and Their Differences/
18. Lab 18 Independent Sample and their Differences.

Recommended in the Board of Studies Meeting of Dept. of MAC held on 10th September 2024



Department of Engineering Mathematics and Computing

B. Tech. (First Semester) Micro/Mini Projects (25241109)

Tentative Topics

1. The mouse pointer can be restricted in particular rectangle. The idea is to create a function called **restrictmouse()** which takes four parameters which containing X coordinate and Y coordinate. First point mention the top of the rectangle and the second point mention the bottom of the rectangle. Below are the functions used for the same:

- **initmouse()**: use to initialize mouse.
- **showmouse()**: shows the mouse pointer on the output screen.
- **restrictmouse()**: used to set Horizontal and vertical limit of the mouse pointer by setting the following parameters. **AX = 7** for horizontal and **AY = 8** for vertical.

2. This following program makes use of some sub function, which were already discussed previously, and shows how they can be used to write useful programs like free-hand drawing. Below are the functions used:

- **initmouse()**: use to initialize mouse.
- **showmouse()**: shows mouse pointer on the output screen.
- **hidemouse()**: used to hide mouse while drawing.
- **getmouseposition()**: Fetches current location of the pointer and draw line accordingly.

3. **(The Sieve of Eratosthenes)** A prime integer is any integer greater than 1 that can be divided evenly only by itself and 1. The Sieve of Eratosthenes is a method of finding prime numbers. It works as follows:

a) Create an array with all elements initialized to 1 (true). Array elements with prime subscripts will remain 1. All other array elements will eventually be set to zero.

b) Starting with array subscript 2 (subscript 1 is not prime), every time an array element is found whose value is 1, loop through the remainder of the array and set to zero every element whose subscript is a multiple of the subscript for the element with value 1. For array subscript 2, all elements beyond 2 in the array that are multiples of 2 will be set to zero (subscripts 4, 6, 8, 10, and so on.). For array subscript 3, all elements beyond 3 in the array that are multiples of 3 will be set to zero (subscripts 6, 9, 12, 15, and so on.).

When this process is complete, the array elements that are still set to 1 indicate that the subscript is a prime number. Write a program that uses an array of 1000 elements to determine and print the prime numbers between 1 and 999. Ignore element 0 of the array.

4. **(Airline Reservations System)** A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system. You'll write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your program should display the following menu of alternatives:

Please type 1 for "first class"

Please type 2 for "economy"

If the person types 1, then your program should assign a seat in the first class section (seats 1–5). If the person types 2, then your program should assign a seat in the economy section (seats 6–10). Your program should then print a boarding pass indicating the person's seat number and whether it's in the first class or economy section of the plane.

Use a single-subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding element of the array to 1 to indicate that the seat is no longer available.



Department of Engineering Mathematics and Computing

Your program should, of course, never assign a seat that has already been assigned. When the first class section is full, your program should ask the person if it's acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message "Next flight leaves in 3 hours."

5. (Total Sales) Use a double-subscripted array to solve the following problem. A company has four salespeople (1 to 4) who sell five different products (1 to 5). Once a day, each salesperson passes in a slip for each different type of product sold. Each slip contains:

- The salesperson number
- The product number
- C++) The total dollar value of that product sold that day

Thus, each salesperson passes in between 0 and 5 sales slips per day. Assume that the information from all of the slips for last month is available. Write a program that will read all this information for last month's sales and summarize the total sales by salesperson by product. All totals should be stored in the double-subscripted array sales. After processing all the information for last month, print the results in tabular format with each of the columns representing a particular salesperson and each of the rows representing a particular product. Cross total each row to get the total sales of each product for last month; cross total each column to get the total sales by salesperson for last month. Your tabular printout should include these cross totals to the right of the totaled rows and to the bottom of the totaled columns.

6. Missing number in array: Given an array of size N-1 such that it only contains distinct integers in the range of 1 to N. Display missing element. Complete the function MissingNumber() that takes array and N as input parameters and returns the value of the missing number.

Input:
N = 5
A[] = {1,2,3,5}
Output: 4

7. Leaders in an Array: Given an array A of positive integers. Your task is to find the leaders in the array. An element of array is leader if it is greater than or equal to all the elements to its right side. The rightmost element is always a leader.

The task is to complete the function leader() which takes array A and n as input parameters and returns an array of leaders in order of their appearance.

Input:
n = 6
A[] = {16,17,4,3,5,2}
Output: 17 5 2

Explanation: The first leader is 17 as it is greater than all the elements to its right. Similarly, the next leader is 5. The right most element is always a leader so it is also included.

8. Kth Smallest Element: Given an array arr[] and an integer K where K is smaller than size of array, the task is to find the Kth smallest element in the given array. It is given that all array elements are distinct.



Department of Engineering Mathematics and Computing

Your task is to complete the function `kthSmallest()` which takes the array `arr[]`, integers `l` and `r` denoting the starting and ending index of the array and an integer `K` as input and returns the `K`th smallest element.

Input:

`N = 6`

`arr[] = 7 10 4 3 20 15`

`K = 3`

Output : 7

Explanation :

3rd smallest element in the given

array is 7.

9. Majority Element: Given an array `A` of `N` elements. Find the majority element in the array. A majority element in an array `A` of size `N` is an element that appears more than $N/2$ times in the array. The task is to complete the function `majorityElement()` which returns the majority element in the array. If no majority exists, return -1.

Input:

`N = 5`

`A[] = {3,1,3,3,2}`

Output:

3

Explanation:

Since, 3 is present more than $N/2$ times, so it is the majority element.

10. Minimum Number of Jumps: Given an array of `N` integers `arr[]` where each element represents the maximum length of the jump that can be made forward from that element. This means if `arr[i] = x`, then we can jump any distance `y` such that $y \leq x$.

Find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is 0, then you cannot move through that element.

Note: Return -1 if you can't reach the end of the array.

Your task is to complete function `minJumps()` which takes the array `arr` and its size `N` as input parameters and returns the minimum number of jumps. If not possible return -1.

Input:

`N = 11`

`arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}`

Output: 3

Explanation:

First jump from 1st element to 2nd element with value 3. Now, from here we jump to 5th element with value 9, and from here we will jump to the last.

11. Equilibrium Point: Given an array `A` of `n` positive numbers. The task is to find the first Equilibrium Point in an array. Equilibrium Point in an array is a position such that the sum of elements before it is equal to the sum of elements after



Department of Engineering Mathematics and Computing

it. The task is to complete the function `equilibriumPoint()` which takes the array and `n` as input parameters and returns the point of equilibrium. Return -1 if no such point exists.

Note: Return the index of Equilibrium point.

Input:

`n = 5`

`A[] = {1,3,5,2,2}`

Output: 3

Explanation:

equilibrium point is at position 3

as elements before it $(1+3) =$

elements after it $(2+2)$.

12. Inversion Count in an Array: Given an array of integers. Find the Inversion Count in the array. Inversion Count: For an array, inversion count indicates how far (or close) the array is from being sorted. If array is already sorted then the inversion count is 0. If an array is sorted in the reverse order then the inversion count is the maximum. Formally, two elements $a[i]$ and $a[j]$ form an inversion if $a[i] > a[j]$ and $i < j$.

Your task is to complete the function `inversionCount()` which takes the array `arr[]` and the size of the array as inputs and returns the inversion count of the given array.

Input: `N = 5, arr[] = {2, 4, 1, 3, 5}`

Output: 3

Explanation: The sequence 2, 4, 1, 3, 5

has three inversions (2, 1), (4, 1), (4, 3).

13. Trapping Rain Water: Given an array `arr[]` of `N` non-negative integers representing the height of blocks. If width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.

Input:

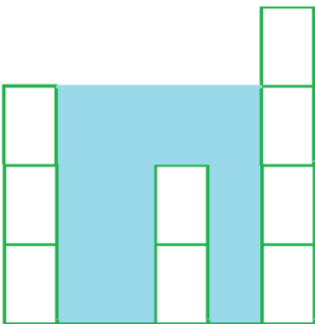
`N = 6`

`arr[] = {3,0,0,2,0,4}`

Output:

10

Explanation:



Bars for input {3, 0, 0, 2, 0, 4}

Total trapped water = $3 + 3 + 1 + 3 = 10$

The task is to complete the function `trappingWater()` which takes `arr[]` and `N` as input parameters and returns the total amount of water that can be trapped.



Department of Engineering Mathematics and Computing

14. Minimum Platforms: Given arrival and departure times of all trains that reach a railway station. Find the minimum number of platforms required for the railway station so that no train is kept waiting. Consider that all the trains arrive on the same day and leave on the same day. Arrival and departure time can never be the same for a train but we can have arrival time of one train equal to departure time of the other. At any given instance of time, same platform can not be used for both departure of a train and arrival of another train. In such cases, we need different platforms. Your task is to complete the function findPlatform() which takes the array arr[] (denoting the arrival times), array dep[] (denoting the departure times) and the size of the array as inputs and returns the minimum number of platforms required at the railway station such that no train waits.

Note: Time intervals are in the 24-hour format(HHMM) , where the first two characters represent hour (between 00 to 23) and the last two characters represent minutes (this may be > 59).

Example 1:

Input: n = 6

arr[] = {0900, 0940, 0950, 1100, 1500, 1800}

dep[] = {0910, 1200, 1120, 1130, 1900, 2000}

Output: 3

Explanation:

Minimum 3 platforms are required to safely arrive and depart all trains.

15. Next Greater Element in an Array: Given an array arr[] of size N having elements, the task is to find the next greater element for each element of the array in order of their appearance in the array. Next greater element of an element in the array is the nearest element on the right which is greater than the current element.

If there does not exist next greater of current element, then next greater element for current element is -1. For example, next greater of the last element is always -1.

Input:

N = 4, arr[] = [1 3 2 4]

Output:

3 4 4 -1

Explanation:

In the array, the next larger element to 1 is 3 , 3 is 4 , 2 is 4 and for 4 ? since it doesn't exist, it is -1.

You only need to complete the function next LargerElement() that takes list of integers arr[] and N as input parameters and returns list of integers of length N denoting the next greater elements for all the corresponding elements in the input array.



Department of Engineering Mathematics and Computing

B. Tech. (First Semester)

Language Lab (25241110)

Course Objectives:

- The course intends to build the required communication skills of the students to communicate effectively in real-life situations like starting a talk and be comfortable using English language.
- It aims at teaching students to appreciate English language through the study of scientific, creative, and academic text.
- The course is designed to acquaint students with structure of English language used in literature, functional varieties, figurative language, and verbal concomitance.
- The students are expected to enrich their knowledge of language, culture, and ethics through this course.

UNIT 1: Communication: Approaches, Elements, Verbal and Nonverbal Communication; Barriers to Communication; Johari Communication Window.

UNIT 2: Listening: Factors Affecting Listening and Improving Listening.

UNIT 3: Public Speaking & Delivering Presentation.

UNIT 4: Reading Passages & Comprehension: Steps and Methods.

UNIT 5: Writing: Essentials of good writing; Drafting CV/biodata/Résumé

Course Outcomes

After successful completion of the course the student will be able to:

| CO's | Description of CO's |
|------|--|
| CO1 | Speak clearly effectively and appropriately in a public forum to a variety of audiences and purposes |
| CO2 | Prepare oral dialogues and arguments within the Engineering Profession effectively |
| CO3 | Demonstrate knowledge and comprehension of major text and traditions in language as well as its social, cultural, and historical context |
| CO4 | Read a variety of Text analytically to demonstrate in writing and/or speech the interpretation of texts. |
| CO5 | Interpret text written in English assessing the results in written and oral arguments using appropriate material for support |

Reference Books: -

- Understanding Human Communication — By Ronald Alderman by OUP
- Communication Skills for Engineers — Pearson Education.
- Practical English Grammar by Thomson Martinet — Oxford University Press
- A Handbook of Language laboratory by P Sreekumar — Cambridge University Press.

*Reading Material for story and poetry is to be selected by concerned teacher in class.

Language Laboratory:

The objective of the language lab is to expose students to a variety of listening and speaking drills. This would especially benefit students who are deficient in English and it also aims at confidence building for interviews and competitive examinations. The Lab is to cover following syllabus.

1. Communication lab exercises as specified in Lab Manual
2. Listening skills (using Marc Hancock, CUP).
3. Speaking skills
4. Oral presentation.

Recommended in the Board of Studies Meeting of Dept. of MAC held on 10th September 2024