### III – Semester (II – B.Tech. – I - Semester)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Subject</th>
<th>POs</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ESC-210</td>
<td>Discrete Mathematics</td>
<td>1,2,3</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ESC-211</td>
<td>Digital Logic Design</td>
<td>1,2,3,4,6</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CS-PCC-211</td>
<td>Computer Organization</td>
<td>1,2,3</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CS-PCC-212</td>
<td>OOP through Java</td>
<td>1,2,3</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CS-PCC-213</td>
<td>Database Management</td>
<td>1,2,3</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CS-PCC-214</td>
<td>Computer Organization</td>
<td>2,3,4,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CS-PCC-215</td>
<td>OOP through Java Lab</td>
<td>2,3,4,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CS-PCC-216</td>
<td>Database Management</td>
<td>2,3,4,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ESC-212</td>
<td>Python Programming</td>
<td>1,2,3,4,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td></td>
<td>15 - 12 21</td>
<td></td>
</tr>
</tbody>
</table>

**Mandatory Course (Non-Credit)**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Subject</th>
<th>POs</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>MC-201</td>
<td>Gender Sensitization</td>
<td>1,6,8,9</td>
<td>- - 3 -</td>
<td></td>
</tr>
</tbody>
</table>

### IV – Semester (II – B.Tech. – II - Semester)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Subject</th>
<th>POs</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BSC-201</td>
<td>Numerical and Statistical Methods</td>
<td>1,2</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CS-PCC-221</td>
<td>Formal Languages and Automata</td>
<td>1,2,3</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CS-PCC-222</td>
<td>Software Engineering</td>
<td>1,2,3,4,6</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CS-PCC-223</td>
<td>Operating Systems</td>
<td>1,2,3</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CS-PCC-224</td>
<td>Computer Networks</td>
<td>1,2,3,6</td>
<td>3 - - - 3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CS-PCC-225</td>
<td>Operating Systems (Linux) Lab</td>
<td>1,3,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CS-PCC-226</td>
<td>Computer Networks Lab</td>
<td>2,4,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CS-PCC-227</td>
<td>Internet of Things Lab</td>
<td>1,2,3,4,5,6</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>BSC-203</td>
<td>Computational Mathematics Lab</td>
<td>3,4,5</td>
<td>- - 3 1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td></td>
<td>15 - 12 21</td>
<td></td>
</tr>
</tbody>
</table>

**Mandatory Course (Non-Credit)**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Subject</th>
<th>POs</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>MC-202</td>
<td>Environmental Sciences</td>
<td>1,6,7</td>
<td>3 - - - -</td>
<td></td>
</tr>
</tbody>
</table>

Note: Summer Internship – I (Mandatory Course) carried out during Summer Vacation between IV semester & V semester and evaluated in V semester.
II-B.TECH.-II-SEMESTER SYLLABUS
NUMERICAL AND STATISTICAL METHODS

II-B.Tech.-II-Sem.
Subject Code: BSC-201

Course Outcomes: Upon completion of the course, the students will be able to
1. solve transcendental, linear and non-linear system of equations using numerical methods
2. find the numerical solutions for first order initial value problems and integrals
3. differentiate among random variables involved in the probability models
4. test hypothesis for small and large samples
5. identify the correlation coefficients, strength, direction and significance level

Unit-I: Algebraic and transcendental Equations and Curve Fitting 9 hours
Curve Fitting: Fitting a linear, second degree, exponential and power curve by method of least squares.

Unit-II: Numerical Integration and Solution of Ordinary Differential Equations 9 hours
Numerical Integration: Trapezoidal rule, Simpson’s 1/3rd and 3/8th rule.
Solution of Ordinary Differential equations: Taylor’s series, Picard’s method of successive approximations, Euler’s method, Runge-Kutta method (second and fourth order)

Unit-III: Probability, Random variables and Distributions (6 + 4) 10 hours
Part A: Probability & Random variables: Random variables, discrete and continuous random variables, probability distribution function, probability density function and mathematical expectations.
Part B: Distributions: Binomial, Poisson and Normal distributions.

Unit – IV: Sampling Theory and Test of Hypothesis for Large Samples 12 hours
Sampling Theory: Introduction, Population and samples, Sampling distribution of means and variances
Test of Hypothesis For Large Samples : Introduction, Hypothesis, Null and Alternative Hypothesis, Type I and Type II errors, Level of significance, One tail and two-tail tests, Tests concerning one mean and proportion, two means-proportions and their differences. Point estimation, Maximum error of estimate and Interval estimation.

Unit – V: Test of Hypothesis for Small Samples 8 hours
Test of Hypothesis for Small Samples: t - Test, F-Test and χ²- Test for goodness of fit and independence of attribute. Point estimation, maximum error of estimate and Interval estimation. Correlation and regression-Rank Correlation.

Textbooks:
1. Introductory Methods of Numerical Analysis by S. S. Sastry, PHI Learning Pvt. Ltd.

References:
FORMAL LANGUAGES AND AUTOMATA THEORY

II-B.Tech.-II-Sem. L T P C
Subject Code: CS-PCC-221  3 - - 3

Course Outcomes: Upon completion of the course, the students will be able to

1. explain the concepts of formal languages and finite automata techniques
2. design various finite automata and its conversion
3. build finite automata for different regular expressions and languages
4. summarize context free grammar and construction of PDA
5. construct turing machines and analyze undecidability

Unit-I 11 hours


Unit-II 9 hours

Finite Automata: Introduction, Deterministic Finite Automata (DFA), Design of DFAs, Non Deterministic Finite Automata (NFA), Non-Deterministic Automata with Є-moves, Design of NFA- Є’s, NFA Versus DFA.

Equivalent Automata: Equivalent Finite-State Automata, Equivalence of NFA/NFA- Є and DFA, Equivalence of NFA, with Є moves to NFA, without Є – moves, Minimization of Finite Automata

Unit-III (5 + 5) 10 hours

Part-A: Regular Expressions and Languages: Regular languages, Regular expressions, Components of Regular Expression, Properties of Regular Expressions.


Unit-IV 10 hours


Unit-V 9 hours

Turing Machine: Introduction, Components of Turing Machine, Elements of TM, Moves of a TM, Language accepted by a TM , Design of TM’s, Types of Turing machines (proofs not required), Chomsky hierarchy of Languages, undecidable Problems, NP and P Problems.

Text Books:
1. Introduction to Automata Theory Languages and Computation, Hopcroft H.E. and Ullman J. D. Pearson Education.

References:
SOFTWARE ENGINEERING

II-B.Tech.-II-Sem. 

Subject Code: CS-PCC-222

Course Outcomes: Upon completion of the course, the students will be able to

1. apply software engineering principles and techniques
2. identify requirements, analyze and prepare models
3. design a system, component or process to meet the desired needs
4. analyze various testing techniques by using various metrics
5. adapt risk management strategies to assure software quality

<table>
<thead>
<tr>
<th>Unit-I</th>
<th>10 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction to Software Engineering:</strong> Evolving role of Software, Changing nature of Software, Software Myths.</td>
<td></td>
</tr>
<tr>
<td><strong>A Generic View Of Process:</strong> Software engineering-A layered technology, The Capability Maturity Model Integration (CMMI), Process Assessment.</td>
<td></td>
</tr>
<tr>
<td><strong>Process Models:</strong> The waterfall model, incremental process models, evolutionary process models, specialized process models, the unified process.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit-II</th>
<th>9 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software Requirements:</strong> Functional and Non functional requirements, User requirements, System requirements, the software requirements document.</td>
<td></td>
</tr>
<tr>
<td><strong>Requirements Engineering Process:</strong> Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.</td>
<td></td>
</tr>
<tr>
<td><strong>System models:</strong> context models, behavior models, data models, object models, structured methods.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit-III</th>
<th>(5 + 5) 10 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part-A: Design engineering:</strong> Design process and design quality, design concepts, the design model, Creating an Architectural Design: Software architecture, data design, architectural styles and patterns, architectural design.</td>
<td></td>
</tr>
<tr>
<td><strong>Part-B: Modeling component-level design &amp; performing user interface design:</strong> Designing Class based components, conducting component level design, Golden rules, user interface analysis and design.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit-IV</th>
<th>10 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Testing Strategies:</strong> A strategic approach to software testing, strategies for conventional software, Black-Box and White-Box testing, Validation Testing, System Testing, the art of Debugging.</td>
<td></td>
</tr>
<tr>
<td><strong>Product Metrics:</strong> Software Quality, Metrics for analysis model, Metrics for design model, Metrics for source code, Metrics for testing, Metrics for maintenance.</td>
<td></td>
</tr>
<tr>
<td><strong>Process and products Metrics:</strong> Software measurement, Metrics for software quality.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit-V</th>
<th>9 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk Analysis and Management:</strong> Risk Management, Reactive vs Proactive risk strategies, Software risks, Risk identification, Risk projection Risk refinement, RMMM, RMMM plan.</td>
<td></td>
</tr>
<tr>
<td><strong>Software Quality Assurance:</strong> Quality Management, Quality concepts, Software quality assurance, Software reviews, Formal technical reviews, Statistical Software Quality Assurance, Software reliability, ISO 9000Quality standards.</td>
<td></td>
</tr>
</tbody>
</table>

| Text Books: |

| References: |
OPERATING SYSTEMS

II-B.Tech.-II-Sem.

Subject Code: CS-PCC-223

Course Outcomes: Upon completion of the course, the students will be able to

1. outline various concepts and structures of operating systems
2. solve synchronization problems by using process management
3. adapt various deadlock handling and memory management mechanism
4. analyze various file management system
5. make use of I/O Management and security mechanisms

Unit-I


Operating Systems Structures: Operating system services and systems calls, system programs, operating system structure.

Unit-II


Unit-III

Part-A: Deadlocks: System model, deadlock characterization, deadlock prevention, detection and avoidance, recovery from deadlock banker’s algorithm.

Part-B: Memory Management: Basic concepts, swapping, contiguous memory allocation, paging, structure of the page table, segmentation, virtual memory, demand paging, page-replacement algorithms, thrashing.

Unit-IV

File Management System: Concept of a file, access methods, directory structure, file system mounting, file sharing, protection. File system implementation: file system structure, file system implementation, directory implementation, allocation methods, free-space management, efficiency and performance.

Unit-V

I/O Management System: Mass storage structure - overview of mass storage structure, disk structure, disk attachment, disk scheduling algorithms, swap space management, stable storage implementation, tertiary storage structure.


Text Books:

References:
2. Operating System a Design Approach-Crowley, TMH.
COMPUTER NETWORKS

II-B.Tech.-II-Sem.

Subject Code: CS-PCC-224

Course Outcomes: Upon completion of the course, the students will be able to

1. outline the basics of computer networks and various layers
2. demonstrate multiple access protocols
3. interpret network layer and routing algorithms
4. illustrate internetworking and various transport protocols
5. make use of various protocols of application layer

Unit-I


Physical layer: Transmission Media, Guided Media, wireless transmission Media.

Data link layer: Design issues, CRC Codes, Elementary Data Link layer Protocols, sliding Window Protocol.

Unit-II

Multiple Access protocols-Aloha, CSMA, Collision free protocols, Ethernet –Physical layer, Ethernet Mac sub layer, Data link layer switching and use of bridges, learning bridges, Spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.

Unit – III


Part-B: Routing Algorithms: Optimality principle, shortest path, flooding, distance vector routing, count to infinity problem, hierarchical routing, congestion control algorithms and admission control.

Unit – IV

Internetworking: Tunneling, internetwork Routing, Packet fragmentation, IPV4, IPV6 Protocol, IP addresses, CIDR, ICMP, ARP, RARP, DHCP.

Transport Layer: Services provided to the upper layers elements of transport protocol-addressing connection establishment, connection release.

Unit-V


Application Layer: Introduction, Providing services, Applications layer paradigms, HTTP, FTP, electronic mail, DNS, SSH.

Text Books:

References:
1. Introduction to Data communication and Networking, Tamasi, Pearson Education
OPERATING SYSTEMS (Linux) LAB

II-B.Tech.-II-Sem.  
Subject Code: CS-PCC-225

Course Outcomes: Upon completion of the course, the students will be able to

1. interpret various CPU scheduling algorithms and file allocation methods
2. experiment with File organization and memory management
3. distinguish Deadlock Avoidance and Deadlock Prevention
4. compare different page replacement and disk scheduling techniques
5. design and develop solutions for using system calls and implementing IPCs

LIST OF EXPERIMENTS

Week 1:
Write C programs to simulate the following CPU Scheduling algorithms:
   a) FCFS  
   b) priority

Week 2:
Write C programs to simulate the following CPU Scheduling algorithms:
   a) SJF  
   b) Round Robin

Week 3:
Write a C program to simulate Bankers Algorithm for Deadlock Avoidance

Week 4:
Write a C program to simulate Bankers Algorithm for Deadlock Prevention

Week 5:
Write C programs to simulate the following memory management techniques:
   a) Fixed Memory Technique (MFT)  
   b) Variable Memory Technique (MVT)

Week 6:
Write C program to simulate the following contiguous memory allocation techniques
   a) First-fit  
   b) Best-fit  
   c) Worst-fit

Week 7:
Write C programs to simulate the following memory management techniques:
   a) Paging  
   b) Segmentation

Week 8:
Write C programs to simulate the following Page Replacement Techniques:
   a) FIFO  
   b) LRU  
   c) Optimal

Week 9:
Write C programs to simulate the following file allocation strategies:
   a) Sequential  
   b) Linked  
   c) Indexed

Week 10:
Write C programs to simulate the following file organization Techniques:
   a) Single level  
   b) Two level  
   c) Hierarchical
Week 11:

Write C program to simulate the following Disk Scheduling algorithms:
a) FCFS  b) SSTF  c) SCAN

Weeks 12:

Write C programs to simulate the following Disk scheduling algorithms:
a) C-SCAN  b) LOOK  c) C-LOOK

Micro-Projects: Student must submit a report on one of the following Micro–Projects before commencement of second internal examination.

1. Multi-level queue CPU scheduling algorithm
2. Producer-consumer problem using semaphore
3. Dining- Philosopher problem using semaphore
4. Multithreading using pthread library
5. Process / Thread synchronization
6. DAG (Directed Acyclic Graph) file organization technique
7. A slower file system mechanism
8. Demand Paging technique of memory management
9. Threaded Matrix Multiply
10. Virtual Memory Simulation

References:

COMPUTER NETWORKS LAB

II-B.Tech.-II-Sem.  
Subject Code: CS-PCC-226

Course Outcomes: Upon completion of the course, the students will be able to

1. make use of NS2/NS3 tools in computer networks
2. outline the concepts of network models and components
3. Adapt various data link layer algorithms and protocols
4. illustrate various network layer algorithms and protocols
5. demonstrate various transport layer algorithms and protocols

<table>
<thead>
<tr>
<th>CO - PO Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
</tr>
<tr>
<td>CO1</td>
</tr>
<tr>
<td>CO2</td>
</tr>
<tr>
<td>CO3</td>
</tr>
<tr>
<td>CO4</td>
</tr>
<tr>
<td>CO5</td>
</tr>
</tbody>
</table>

3-Strong; 2-Medium; 1-Weak

LIST OF EXPERIMENTS

1. Familiarization with Networking Components and devices (LAN Adapters, Hubs, Switches, Routers, Co-axial cable, Crimping Tool, Connectors etc.)
2. Implement the data link layer framing method using character stuffing and bit stuffing.
3. Implement CRC on a data set of characters using CRC-12 / CRC-16 polynomial.
4. Implement Stop and Wait Protocol
5. Implement Sliding Window Protocol.
6. Implement Dijkstra's shortest path algorithm through a graph.
7. Obtain Routing table at each node using distance vector routing algorithm for a given subnet graph with weights indicating delay between nodes.
8. Implement a Hierarchical routing algorithm.
9. Obtain broadcast tree for given subnet of hosts.
10. Implement collision free protocol.
11. Simulate ARP / RARP protocols using NS2/NS3 tools.
12. Implement the token bucket congestion control algorithm.

Micro-Projects: Student must submit a report on one of the following Micro–Projects before commencement of second internal examination.

1. Peer to Peer File Sharing Technology over LANs.
6. Network Administrator Tool Project.
7. Domain Name Service with Secured Manager.

Reference:

INTERNET OF THINGS LAB

II-B.Tech.-II-Sem.                      L  T  P  C
Subject Code: CS-PCC-227                 0  0  3  1.5

Course Outcomes: Upon completion of the course, the students will be able to

1. improve working on basic IoT devices
2. determine learning and utilization of IoT devices
3. develop automation work-flow in IoT enabled environment
4. recommend working on advance IoT Systems
5. take part in practicing and monitoring remotely

List of Experiments (Minimum 10 experiments to be conducted)

1. Familiarization with Arduino/Raspberry Pi and perform necessary software installation.
2. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
3. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
4. To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.
5. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
6. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
7. To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smart phone using Bluetooth.
8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when ‘1’/’0’ is received from smart phone using Bluetooth.
9. Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to cloud.
10. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from cloud.
11. Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker.
12. Write a program on Arduino/Raspberry Pi to subscribe to MQTT broker for temperature data and print it.
13. Write a program to create TCP server on Arduino/Raspberry Pi and respond with humidity data to TCP client when requested.
14. Write a program to create UDP server on Arduino/Raspberry Pi and respond with humidity data to UDP client when requested.

Micro-Projects: Student must submit a report on one of the following Micro–Projects before commencement of second internal examination.

1. Air Pollution Meter
2. Smart Garbage Collector
3. SMART Garage Door
4. Humidity & Temperature Monitoring
5. Baggage Tracker
6. Smart Trash Collector
7. Liquid Level Monitor
8. Circuit Breakage Detection
9. Human Safety Night Patrolling IOT Project
10. Anti-Theft Flooring System

Reference:

1. Internet of Things Lab Manual, Department of CSE, CMRIT, Hyd.
Course Outcomes: After completion of this course using Sci LAB / MATLAB, students will be able to

1. solve problems on Algebraic, Matrices and Calculus
2. find roots of an equation using different Methods
3. solve interpolation problems
4. fit a curve for straight line, parabola, exponential and power curves
5. evaluate Numerical differentiation and integration

CO – PO Mapping

<table>
<thead>
<tr>
<th>POs</th>
<th>CO1</th>
<th>CO2</th>
<th>CO3</th>
<th>CO4</th>
<th>CO5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>PO2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>PO3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

- 3-Strong; 2-Medium; 1-Weak

LIST OF EXPERIMENTS

Week-1: Basic Features

a. To know the history and features of Sci LAB / MATLAB
b. To know the local environment of Sci LAB / MATLAB

Week-2: Algebra

a. Find the values of x, y, z of the equations x+y+z=3, x+2y+3z=4, x+4y+9z=6
b. For \( f(x) = 8x^8 - 7x^7 + 12x^6 - 5x^5 + 8x^4 + 13x^3 - 12x + 9 \) compute \( f(2) \), roots of \( f(x) \) and plot the graph

Week-3: Matrices

a. Find the addition, subtraction and multiplication of matrix

\[
A = \begin{bmatrix}
1 & 2 & 9 \\
2 & 1 & 2 \\
3 & 4 & 3 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
1 & 2 & 5 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix}
\]

b. Find the transpose of matrix \( A = \begin{bmatrix}
1 & 2 & 9 \\
2 & 1 & 2 \\
3 & 4 & 3 \\
\end{bmatrix} \)

c. Find the inverse of matrix \( A = \begin{bmatrix}
1 & 2 & 3 \\
2 & 3 & 2 \\
1 & 2 & 5 \\
\end{bmatrix} \)

Week-4: Linear Transformation

a. Find the characteristics equation of the matrix

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 0 \\
\end{bmatrix}
\]

b. Find the Eigen values of the matrix

\[
\begin{bmatrix}
1 & 8 & 10 \\
4 & 2 & 4 \\
5 & 2 & 8 \\
\end{bmatrix}
\]
c. Find the Eigen vector of the matrix
\[
\begin{bmatrix}
3 & 1 & 1 \\
1 & 0 & 2 \\
1 & 2 & 0
\end{bmatrix}
\]

Week-5: Differentiation and Integration

a. Solve \((D^2 + 5D + 6)y = e^x\)

b. Solve \(\int \int x(x^2 + y^2)dx\,dy\)

c. Solve \(\int \int \int xyz\,dx\,dy\,dz\)

Week-6: Root Finding

Write a program to find the root of the equation \(x^3 - 5x + 3 = 0\) by using

a) Bisection Method
b) Regula Falsi

Week-7: Root Finding

Write a program to find the root of the equation \(x^3 - x - 10 = 0\) by using

a) Iteration Method
b) Newton Raphson Method

Week-8: Interpolation

a. Find the Lagrange’s polynomial for the following data \((0,2),(1,3),(2,12),(5,147)\)
b. Fit a straight line for the following data \((0,12),(5,15),(10,17),(15,22),(20,24),(25,30)\)
c. Fit a polynomial curve for the following data \((0,1),(1,1.8),(2,1.3),(3,2.5),(4,6.3)\)

Week-9: Curve Fitting

a. Write a program to find a line of best fit from the given two arrays of \(x\) and \(y\) of same size.
b. Write a program to find a curve of the form \(y = Ax^2 + Bx + C\) the given two arrays of \(x\) and \(y\) of same size.

Week-10: Power Curves

a. Write a program to find a curve of the form \(y = Ae^{bx}\) from the given two arrays of \(x\) and \(y\) of same size.
b. Write a program to find a curve of the form \(y = Ax^b\) the given two arrays of \(x\) and \(y\) of same size.

Week-11: Numerical Differentiation and Integration

a. Write a program to evaluate definite integral using trapezoidal rule, Simpsons 1/3rd rule and 3/8th rule.
b. Write a program to solve to given differential equation using Taylor’s series.

Week-12: Numerical Differentiation and Integration

a. Write a program to solve to given differential equation using Picards and Euler’s method.
b. Write a program to solve to given differential equation using Runge - Kutta method (2nd and 4th Order).
Micro-Projects: Student must submit a report on one of the following Micro–Projects before commencement of second internal examination.

1. Demonstrate the battery discharge function graphically by adopting a mathematical model.
2. Apply inverse Laplace transforms in image processing for getting the better image.
3. Evaluate the trigonometric functions using Laplace transforms.
4. Illustrate the laminar flow of heat through partial differential equations.
5. Design a mathematical model to explain the functioning of Global positioning system (GPS)
6. Design a mathematical model for the construction of flyover
7. Model any art craft using mathematical calculations (electrical / non-electrical)
8. Prepare a detailed report on usage of mathematical concepts in overcoming “risk vs reward” situations in day to day life.
9. 2-D plotting using SCI-lab.
10. 3-D plotting using SCI-lab.

Reference:

II-B.Tech.-II-Sem.
Subject Code: MC-202

Course outcomes: Upon completion of the course, the student will be able to
1. identify the importance, scope and role of ecosystem in our lives
2. interpret nature of available resources and choose an inter-disciplinary approach to environmental protection
3. outline bio-diversity and its relevance to ecological balance
4. explain laws and legislations on environmental protection
5. evaluate technologies for achieving sustainable development

Unit I: Ecosystem
Introduction to ecosystem: Definition, Scope and Importance; Classification of ecosystem; Structure and functions of ecosystem food chain food web, ecological energetic, eco-pyramids, carrying capacity; Biogeochemical cycles (Carbon and Nitrogen Cycles), flow of energy; Environment movement in India (Medha Patkar, Sundarlal Bahuguna, Indira Gandhi, Rachael Carson).

Unit II: Natural Resources
Renewable and Non–renewable resources–Importance, uses, classification of natural resources
(i) forest: deforestation, timber extraction & conservation (ii) water: conflicts over water, dams – benefits & effects; use and over exploitation of water resources , floods, droughts (iii) mineral : use and exploitation, effects on mining, (iv) energy resources: growing needs, renewable and non renewable energy sources, use of alternative energy (v) land resources: land degradation, landslides, soil erosion and desertification; role of an individual in conservation of natural resources and equitable use.

Unit III: Biodiversity
Part-A: Definition and levels of biodiversity, Values of biodiversity Bio– geographical classification of India; hot spots of biodiversity; India as a mega diversity nation; Threats to biodiversity; Endangered and endemic species of India.

Unit IV: Environmental Pollution & Control Technologies

Unit V: Environmental Acts, EIA & Sustainable Development
Textbooks:

References:
1. Environmental Science and Technology by M. Anji Reddy(2007), B.S Publications,