Institute of Infrastructure Technology Research and Management

B.Tech. In Computer Engineering

Semester 5

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243004</td></tr>
<tr><td>II</td><td>Course Title</td><td>Computer Networking</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>3</td><td>4.5</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>None</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
• Understand various aspects of networked system design.
• Understand the existing protocol stacks and interactions among protocols.
Design and develop networked applications using socket programming.</td></tr>
<tr><td>VI</td><td>Course Content</td><td><b>Module 1:</b> Introduction to Computer networks and Internet, End systems, communication channels and interconnecting devices, Performance parameters: throughput, delay, etc., Layered Architecture (OSI and TCP/IP); Applications: Network application Design, Socket Programming, Client-server applications, WWW, Email, FTP, DNS, Peer to Peer file sharing applications.

<b>Module 2:</b> Transport Layer: End to end delivery issues, Reliable data transfers, Congestion Control, Traffic engineering and Quality of service, Flow control and Leaky bucket algorithm, TCP and UDP protocol specification.

<b>Module 3:</b> Internetworking and Routing: Best effort Service, Virtual Circuits, IP Addressing; Routing Issues, Distance Vector and Link State routing, Intra and Inter Autonomous System Routing (OSPF, RIP, BGP); Software defined networking: Control plane and data plane, open flow protocol stack; Unicast, Anycast, Broadcast and Multicast Routing.

<b>Module 4:</b> Data link layer: Introduction, Media access protocols (ALOHA, CSMA based), Ethernet 802.3, Token Bus (802.4) and Token ring (802.5); Reliability Issues; Introduction to Wireless networks: WiFi (802.11), Cellular: GSM and 5G; Introduction to Quality of service and Security over IP, IPV6, Internet of Things (IoT)</td></tr>
<tr><td>VII</td><td><b>Text / Reference books:</b></td><td>1. James Kurose and Keith Rose, “Computer Networking: A Top-Down Approach”, Pearson Education
2. Larry L Peterson and Bruce S Davie, “Computer Networks: A Systems Approach”, Elsevier
3. Andrew S Tanenbaum, “Computer Networks”, Pearson Education
4. Behrouz A Forouzan, “Data Communication and Networking”, McGraw Hill
William Stallings, “Computer networking with Internet protocols and Technology”, Pearson Education</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243002</td></tr>
<tr><td>II</td><td>Course Title</td><td>Operating Systems</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>3</td><td>4.5</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>None</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
• understand the functioning and design of operating systems.
• understand the issues and solutions related to process management, memory management, file system and IO management.
design multi-process and multi-threaded programs, programs using system calls, and shell scripts</td></tr>
<tr><td>VI</td><td>Course Content</td><td><b>Introduction:</b> History and evolution of operating systems; role of an operating system as resource manager; types of operating systems and kernels; system calls.

<b>Process management:</b> creation of process and its address space, system calls such as fork() and exec(); process state transitions, scheduling algorithms and context switching; inter-process communication using shared memory and message passing; process synchronization using locking variables, semaphore, Condition Variables, Barriers etc. ; thread models and multi-threaded programming; resource allocation to processes and deadlock: types of resources, conditions for resource deadlock; algorithms for deadlock detection and recovery, deadlock avoidance, and deadlock prevention;

<b>Memory management:</b> memory hierarchy; virtual memory implementation using paging and segmentation, fragmentation, address translation; memory swapping and page replacement algorithms.

<b>File system and IO management:</b> organization of file system, directory structure, security and access control; direct memory access; disk space management, disk scheduling algorithms, case study and implementation of Unix file system; storage area network – NFS, RAID, etc.

Introduction to virtual machines, operating systems for the resource constrained devices (e.g., Android), energy consumption; Realtime operating systems.</td></tr>
<tr><td>VII</td><td><b>Text / Reference books:</b></td><td>1. Operating System Principles by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, John Wiley publisher
2. Operating System s- Internals and Design Principles by William Stallings, Pearson publisher
3. Modern operating systems by Andrew S Tanenbaum, Pearson publisher
4. Dhamdhere, D. M., Operating Systems - A concept-based approach, McGraw-Hill Education.
The Design of the Unix Operating System by Maurice Bach, Prentice Hall publisher</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243001</td></tr>
<tr><td>II</td><td>Course Title</td><td>Machine Learning</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>3</td><td>4.5</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>None</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
• Understand various aspects of Artificial Intelligence and Machine Learning
• Understand existing algorithms for regression, classification and clustering.
• Design and develop applications using the existing tools and technologies</td></tr>
<tr><td>VI</td><td>Course Content</td><td><b>Module 1:</b> Introduction to Artificial Intelligence and Machine Learning; Applications; Search Algorithms: introduction, breadth first search, depth first search, bidirectional search, A*; Basics of Machine learning: Supervised learning, Unsupervised learning, Semi Supervised learning, Reinforcement learning, etc., Model evaluation and error measures.

<b>Module 2:</b> Regression: Single variable and multivariate Linear Regression, model representation, cost function; Non-linear regression – polynomial regression; Classic algorithms: support vector regression, decision tree regression, random forest regression etc.; Error metrics and measures.

<b>Module 3:</b> Classification: Hypothesis representation, decision boundary, cost function, gradient descent algorithm, overfitting and regularization etc.; Classic algorithms: Logistic regression, Naïve bayes, Decision Tree, Ensemble learning, K Nearest Neighbour, Support Vector Machine; Multiclass classification; Error metrics and measures: precision, recall, scores etc., skewed classes scenario.

<b>Module 4:</b> Artificial Neural Networks: Neurons and brain, Non-linear hypothesis, Model representation, Back Propagation algorithm, ANN for Regression and Classification, Loss function, Hyper parameter tuning; Unsupervised Learning: Clustering algorithms: K-means clustering, Dendrograms and Hierarchical clustering, DBSCAN; Error measures; Dimensionality reduction: Motivation, Principal Component Analysis</td></tr>
<tr><td>VII</td><td><b>Text / Reference books:</b></td><td>1. Machine Learning, Tom Mitchell, MacGrow Hill publisher
2. Artificial Intelligence - A Modern Approach, Stuart Russell, Peter Norvig, Pearson publisher
3. Understanding Machine Learning - From Theory to Algorithms, Shai Shalev-Shwartz and Shai Ben-David, Cambridge University Press</td></tr>
</table>

<table>
<tr><td>I</td><td>Course Code</td><td>CS 243003</td></tr>
<tr><td>II</td><td>Course Title</td><td>Software Engineering</td></tr>
<tr><td>III</td><td>Credit Structure</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>IV</td><td>Prerequisites:</td><td>None</td></tr>
<tr><td>V</td><td>Learning Outcomes:</td><td>After completing this course, the students will be able to
• To understand the best practices in software engineering.
• To develop the necessary skills to handle software projects in a principled way.
• analyze and specify software requirements.
• Apply software engineering principles and techniques to develop large-scale software systems.
• Ability to plan and work effectively in a team.</td></tr>
<tr><td>VI</td><td>Course Content</td><td>1. Introduction of Software Engineering: Need for software engineering, Software Quality attributes, Software product pipelines, Software life cycle models and processes, Requirement engineering using UML Diagrams.
2. Software Architecture and Design: Design principles, Design Patterns, Architecture
3. Versus Design, Modularity, Software Components and Connectors, Architecture Styles.
4. Essence of Modern Software Engineering: Software engineering essence, Essence
language, Essence kernel, Using essence kernel in agile development practices, Agile.
Principles, Agile process models through essence kernel, large scale complex development Using kernel.
5. Software Testing: Quality metrics, Coding style and Static analysis tools, Verification and validation, Various testing techniques, and Test case generations.
6. Software Project Management: Software versioning and Continuous integration, Project management and Risk analysis, Configuration management, Cost analysis and estimation.

Laboratory

Assignment/Project on Software requirement acquisition, UML diagrams, preparing software requirement specification, Practicing agile methods, User story, Backlog, Test case generation, Unit testing, CI configurations, Cost estimation, Manpower management and Sprint analysis using Burn down charts.</td></tr>
<tr><td>VII</td><td><b>Text / Reference books:</b></td><td>1. R.S. PRESSMAN, B.R. MAXIM (2019), Software Engineering: A Practitioner’s
Approach, McGraw-Hill India, 2019, 9th Edition.
2. Mark Richards, Neal Ford (2020), Fundamentals of Software Architecture, O'Reilly
Media, Inc.
3. L. BASS, P. CLEMENTS, R. KAZMAN (2012), Software Architecture in Practice,
Pearson, 3rd Edition.
4. I. JACOBSON, H. LAWSON, P.W. NG, P.E. McMAHON, M. GOEDICKE (2019), The Essentials of Modern Software Engineering, ACM Books.
Ahmed, A., (2011), Software Project Management: A Process-Driven Approach, Auerbach Publications</td></tr>
</table>