Course Description

4th Semester:

Title of Course: Computer Networks **Course Code: 18B11CI411**

L-T-P Scheme: 3-0-0 **Credit: 3**

Objective:

The objective of this course is to build basic concepts of Computer network established for the data communication. This course also aims to provide the fundamental concepts in the design and implementation of networks, their protocols and its applications.

Learning Outcomes:

Computer Networks	
Course Outcome	**Description**
CO1	Outline basics to advanced concepts and techniques of Computer networks.
CO2	Describe problem solving approaches as applied in Data communication networking areas.
CO3	Analyse performance of basic communication networks using both analytical and simulation techniques.
CO4	Develop the Computer network design techniques and practical implementation issues.
CO5	Understand the basic properties of internet and data traffic properties.
CO6	Apply verification and validation techniques on a given software project.
CO7	Demonstrate deployment and basic maintenance skills.

Course Content:

Unit I: Introduction: Introduction to computer network, classification of networks WAN, MAN, LAN), distributed systems, digital signals and data rates, bit stream, symbols and band rate, transmission media, modems, structure of computer network, circuit, packet, message switching, Network topological, Network model, ISO-OSI model, TCP/IP model, primitives and services.

Unit II: Physical Layer: Physical Layer Design Issues (Service provided to data link Layer) Introduction Transmission media, RS-232-C and RS-449, Line coding, Switching Techniques.

Unit III: Data Link Layer: Data Link Layer Design Issues (Service Provided to N/w Layer), Framing, error control, flow control, Link Management, Error Detection and Error Correction Coding, Data Link Protocols (Elementary and sliding Window), local and metropolitan area networks. The Medium Access sub layer, Static and Dynamic Channel Allocation in LANs and MANs, ALOHA Protocols (Pure and Slotted), Different Protocols of LAN, IEEE Standard 802 for LAN (802.2, 802.4, 802.5).

Unit IV: Network Layer: Network Layer Design Issues (Service Provided to Transport Layer). Routing, Congestion, Internetworking. Routing Algorithms, Congestion Control Algorithm Internetworking, congestion control. Design issues, buffer management, synchronization. Session and presentation layer synchronization issues, formatting, data compression, data security.

Unit V: Transport Layer: Transport Layer Design Issue .Connection Management, Buffer Management, Quality of Service. Session Layer Design Issues Synchronization issues. Introduction to Presentation Layer. Encryption and decryption. RSA algorithm.

Teaching Methodology

This course will help the students to facilitate interaction and information transfer over large distances. With internet, computer and telephone networks, buisenesses can allocate their resources efficiently. The Students will be able to learn basic concepts of computer network, its working principle & operation of Internet and Intranet. They will also learn the working principle of operation of LAN, WAN, MAN, congestion in the network and network management.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1, Unit-2
Test-2	25 Marks	Based on Unit-2, 3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-4 and Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Telecommunication networks (will be added from time to time):

Text Books:

1. A.S. Tennenbaum, Computer Networks, PHI
2. W. Stallings, Data & Computer Communication, PHI
3. Forouzen, Behrouz A.Fegan, Sophia Chung Data Communications and Networking,

TMH

Reference Books:

1. Carne, E. Bryan Professional's Guide to Data Communication in a TCP/IP World Artech House, London, 2004
2. Young, Margret Levine Internet: The Complete Reference, Tata McGraw Hill, New Delhi, 2002

Web References:

1. **www.britannica.com**
2. **www.vssut.ac.in**

Journals References:

1. International Journal on Advances in Telecommunications
2. Journal of Network and Computer applications- Elsevier
3. IEEE transactions on networking
4. ACM Journals on networking

Title of Course: Algorithms and Problem Solving **Course Code: 18B11CI412**

L-T-P Scheme: 3-1-0 **Credit: 4**

Objective:

Analysis of common algorithms for processing strings, trees, graphs and networks. Comparison of sorting and searching algorithms. Algorithm design strategies: divide and conquer, dynamic programming, greedy, back tracking, branch and bound. Introduction to NP-completeness.

Learning Outcomes:

Computer Networks +Lab	
Course Outcome	**Description**
CO1	Use critical thinking for problem solving and analyze time complexity of algorithms
CO2	Apply algorithms efficiently and correctly and argue algorithm correctness
CO3	Design efficient algorithms using well-known methods
CO4	Describe effectively, in writing and in an oral presentation, an algorithm and its implementation
CO5	Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline

Course Content:

Unit I: Introduction: Analysis of Algorithm: The efficient algorithm, Average, Best and worst case analysis, Amortized analysis , Asymptotic Notations, Analyzing control statement, Loop invariant and the correctness of the algorithm, Sorting Algorithms and analysis: Bubble sort, Selection sort, Insertion sort, Shell sort Heap sort, Sorting in linear time : Bucket sort, Radix sort and Counting sort

Unit II: Divide and Conquer Algorithm:

Introduction, Recurrence and different methods to solve recurrence, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential**.**

Unit III: Dynamic Programming: Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence

Unit IV Greedy Algorithm: General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm - Activity selection problem, Elements of Greedy Strategy, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem, Job Scheduling Problem, Huffman code

Unit V: Exploring Graphs and Backtracking and Branch and Bound: An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort, Connected components, Introduction, The Eight queens problem , Knapsack problem, Travelling Salesman problem, Minimax principle

Unit VI: Exploring String Matching and NP-Completeness: Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm. The class P and NP, Polynomial reduction, NP- Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Hamiltonian problem.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1, Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 and Unit-6 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides will be added from time to time on N-Drive.

Text Books:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, “Introduction to Algorithms”, Printice Hall of India.

Reference Books:

1. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
2. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006.
3. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997
4. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
5. Harsh Bhasin,”Algorithm Design and Analysis”,First Edition,Oxford University Press.
6. Gilles Brassard and Paul Bratley,Algorithmics:Theory and Practice,Prentice Hall,1995.

Web References:

1. https://onlinecourses.nptel.ac.in/noc23_cs53/preview
2. **https://nptel.ac.in/courses/106105171**

Title of Course: **Operating Systems** **Course Code: 18B11CI413**
L-T-P scheme: 3-0-0 **Credit: 3**

Prerequisite: Students must have knowledge of C programming and working of the computer systems.

Objective:

1. To familiarize with the basic functionality and the evolution of different types of operating systems.
2. To Learn and understand various algorithms related to CPU scheduling, deadlocks, memory management, and storage management.
3. To learn basic aspects of real time operating systems.

Learning Outcomes:

Course Outcome	**Description**
CO1	Gain knowledge of OS fundamentals along with process management concepts
CO2	Apply various process management concepts including scheduling, synchronization, dead-lock to solve given problem.
CO3	Explain various memory management techniques including virtual memory.
CO4	Analyse issues related to memory management.
CO5	Understand file system including disk structure by applying disk scheduling algorithm.
CO6	Work as a team on a project.

Course Content:

Unit-1: Introduction: Operating system structure, Operating system operations, Distributed systems, Special purpose systems, Computing environments, Open source operating systems.

Unit-2: CPU Scheduling: Process concepts: Process states, Process control block, Scheduling queues, Schedulers, Context switch, Multi-threaded programming: Overview, Multithreading models, Threading issues, Process scheduling: Basic concepts, Scheduling criteria, scheduling algorithms.

Unit-3: Synchronization: The Critical section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, monitors.

Unit-4: Deadlocks: Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

Unit-5: Memory management: Memory management strategies, Swapping, Contiguous memory allocation, Paging, Structure of the page table, Segmentation.

Unit-6: Virtual Memory: Demand paging, copy on write, page replacement, allocation of frames, thrashing.

Unit-7: Storage Management: File concept, Access methods, directory structure, file system structure, directory implementation, allocation methods, free space management, disk structure, and disk-scheduling.

Unit-8: Case study on UNIX based Operating system: Design principles, Kernel modules, Process management, Memory management.

Unit-9: Real time systems: Characteristics of Real time operating systems, classification of real time systems, Micro kernels and RTOS, scheduling in RTOS, Rate monotonic scheduling, EDF, Priority inversion

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1, Unit-2 & Unit-3
Test-2	25 Marks	Based on Unit-4 & Unit-5 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-6 to Unit-9 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Operating Systems (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

1. "Operating System Concepts"; A. Silberschatz , P. B. Galvin & G. Gagne , Wiley 10e 2018.
2. "Operating Systems: Internals and Design Principles"; W. Stallings, Pearson 9e, 2017.

Reference Books/Material:

1. "Real time systems design and analysis"; P. A. Laplante & S. J. Ovaska, Wiley, 2013.
2. "Real time systems: Theory and Practice"; Mall R., Pearson, 2e, 2009.

Title of Course: Artificial Intelligence and Applications **Course Code: 18B11CI415**
L-T Scheme: 3-1-0 **Course Credits: 4**

Objectives: In this course we will study the basic components of an intelligent system, their functions, mechanisms, policies and techniques used in their implementation and examples.

Learning Outcomes: The students will have a detailed knowledge of the concepts of artificial intelligence, various applications of AI in different fields, Aware of a variety of approaches to AI techniques.

Course Outcome	Description
CO1	Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
CO2	Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.
CO3	Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing.
CO4	Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning. playing.
CO5	Formulate and solve problems with uncertain information using Bayesian approaches.
CO6	Apply concept Natural Language processing to problems leading to understanding of cognitive computing.

Course Contents:

Unit-1 (Introduction to AI): Definitions, Goals of AI, AI Approaches, AI Techniques, Branches of AI, Applications of AI.
Introduction of Intelligent Systems: Agents and Environments, Good Behavior: the concept of Rationality, The Nature of Environments, The structure of Agents, How the components of agent programs work.

Unit-2 (Problems Solving, Search and Control Strategies)
Solving Problems by Searching, Study and analysis of various searching algorithms. Implementation of Depth-first search, Problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform-cost search, Depth-first search, Depth-limited search, Iterative deepening depth-first search, Bi-directional search Informed (Heuristic) Search Strategies: Greedy best-first search A* search: Minimizing the total estimated solution cost, Conditions for optimality: Admissibility and consistency, Optimality of A*, Memory-bounded heuristic search, Heuristic Functions, Generating admissible heuristics from sub problems: Pattern databases, Learning heuristics from experience.
Beyond Classical Search: Local Search Algorithms and Optimization Problems: Hill-climbing search Simulated annealing, Local beam search, Genetic algorithms, Local Search in Continuous Spaces, Searching with Non-deterministic Actions: AND-OR search trees, Searching with Partial Observations.
Adversarial Search and Constraint Satisfaction Problems, Study of min-max algorithm Adversarial Search: Games, Optimal Decisions in Games, The mini-max algorithm, Optimal decisions in multiplayer games, Alpha--Beta Pruning, Move ordering , Imperfect Real-Time Decisions, Evaluation functions, Cutting off search, Forward pruning, Search versus lookup, Stochastic Games, Evaluation functions for games of chance, Partially Observable Games
Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Variations on the CSP formalism, Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs,

Local Search for CSPs, Alpha-beta pruning and CSP, Implementation aspects of mini-max algorithm and CSP.

Unit- 3 (Knowledge Representations Issues, Predicate Logic, Rules)
Knowledge representation, KR using predicate logic, KR using rules. Reasoning System - Symbolic, Statistical: Reasoning, Symbolic reasoning, Statistical reasoning.

Unit-4 (Quantifying Uncertainty, Learning Systems)
Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Bayes' Rule and Its Use, Representing Knowledge in an Uncertain Domain, Other Approaches to Uncertain Reasoning, Rule-based methods for uncertain reasoning, Representing vagueness: Fuzzy sets and fuzzy logic, Study of fuzzy logic and Decision trees, Implementation aspects of Decision trees.

Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, The decision tree representation, Expressiveness of decision trees, inducing decision trees from examples.

Unit-5 (Expert Systems)
Introduction, Knowledge acquisition, Knowledge base, Working memory, Inference engine, Expert system shells, Explanation, Application of expert systems.

Fundamentals of Neural Networks: Introduction and research history, Model of artificial neuron, Characteristics of neural networks, learning methods in neural networks, Single-layer neural network system, Applications of neural networks.

Fundamentals of Genetic Algorithms: Introduction, Encoding, Operators of genetic algorithm, Basic genetic algorithm.

Evalution Scheme:

Evaluation	Marks	Remarks
T1	15 Marks(1 Hr.)	1st -4th Week
T2	25 Marks(1:30 Hr.)	5th – 10th Week
T3	35 Marks(2:00 Hr.)	11th -16th Week
Tutorial/Presentation	10	
Assignments	5	
Quiz	5	
Attendance	5	
Total	**100**	

Text Books

1. Rich, Elaine Knight, Kevin, Artificial Intelligence, Tata McGraw Hill.
2. Luger, George F, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education.

References

1. Nilsson, Nils J, Artificial Intelligence, Morgan Kaufmann, Russell, Stuart J. Norvig, Peter, AI: A Modern Approach, Pearson Education.
2. Neural networks and Learning Machines, Simon Haykin, PHI Learning Pvt. Ltd.

Title: Algorithms Lab **Code: 18B17CI472**
L-T-P scheme: 0-0-2 **Credit: 1**

Prerequisite: Experience in programming is desirable. Student must have already registered for "Software Development Lab" (18B17CI171) and "Data Structures lab" (18B17CI371).

Objective:

1. To provide exposure to problem-solving through programming.
2. Strengthen higher level cognitive Skills of analysis of problem, creation of solution and evaluation of performance.
3. Strengthen Ability of data abstraction and problem solving using computer
4. Strengthen ability to express solution to problem clearly and precisely.
5. Strengthen ability to design and evaluate ADTs, nonlinear temporary and persistent data structures and also related algorithms.
6. Introduce students to some domain specific data structures and related algorithms in various domains.

Learning Outcomes:

Course Outcome	**Description**
CO1	Design new algorithms, prove them correct, and analyze their asymptotic and absolute runtime and memory demands.
CO2	Find an algorithm to solve the problem (create) and prove that the algorithm solves the problem correctly (validate).
CO3	Understand the mathematical criterion for deciding whether an algorithm is efficient, and know many practically important problems that do not admit any efficient algorithms.
CO4	Apply classical sorting, searching, optimization and graph algorithms.
CO5	Understand basic techniques for designing algorithms, including the techniques of Recursion, Divide-and-Conquer, Greedy Algorithms and Dynamic Programming

Course Content:

The following assignments will be carried out in synchronization with the theory classes.

Unit-1: Development of programs including analysis of algorithm Asymptotic Notation, Sorting and merging Algorithm.

Unit-II: Programs using Heap, Priority Queues, B-Tree, AVL, Splay Tree, Red-Black Tree, Threaded Tree.

Unit-III: Programs using Classification, Record Organization, and Retrieval System of files External Sorting. Design, Analysis, integration of set & dictionary, collision resolution and analysis

Unit-IV: Programs using Divide and Conquer method, Dynamic programming, Introduction to Greedy Method.

Unit-V: Program using String operation, pattern matching algorithm, tries, text compression, text similarity testing application.

Units to Lab Mapping:

Unit	Labs
I	1, 2, 3
II	4, 5
III	6, 7, 8
IV	9, 10, 11
V	12, 13, 14

Teaching Methodology:

This course is introduced to help students understand the designing and analysis of algorithm. Any (C, C++, JAV etc) programming language used to implement algorithms. Starting from the programming environment setup, the student will slowly be exposed to program designing and later to complexity analysis fundamentals. The entire course is broken down into five separate units, from fundamentals of algorithms to some complex algorithms designing methodology like Dynamic Programming Greedy Techniques etc.

Evaluation Scheme:

Exams		Marks	Coverage
P-1		15 Marks	**Based on Lab Exercises: 1-6**
P-2		15 Marks	**Based on Lab Exercises: 7-13**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Study material of Algorithms Lab (will be added time to time): Digital copy will be available on the JUET server.

Text Book:

1. Thomas H., Coremen: Introduction to algorithm, the Massachusetts institute of Technology, Cambridge, Massachusetts.

Reference Books/Material:

1. Aho, Hopcraft, Ullman: Data Structure and Algorithms
2. Kruse, Tonso, Leung: Data Structure and program Design in C
3. Sahani: Data structure and algorithm and application in C++
4. Weiss: Data Structure and Algorithm analysis in C/C++

Online Courses:
NPTEL-Algorithms and Problem Solving: https://nptel.ac.in/courses/106/105/106105164/

Videos Available on YouTube:
https://www.youtube.com/watch?v=OQ5jsbhAv_M

https://www.youtube.com/watch?v=huQojf2tevI
https://www.youtube.com/watch?v=sSno9rV8Rhg

Website

- https://www.geeksforgeeks.org
- https://www.indiabix.com
- https://www.includehelp.com
- https://www.tutorialspoint.com
- https://www.sanfoundry.com
- https://www.programiz.com

Coding Platforms

- https://www.codechef.com
- https://www.hackerrank.com
- https://www.interviewbit.com
- https://www.spoj.com
- https://www.hackerearth.com
- https://leetcode.com

Integrated Development Environment

- Turbo C++
- Dev-c++
- Code::Block

Title of Course: **Operating Systems Lab** **Course Code: 18B17CI473**

L-T-P scheme: 0-0-2 **Credit: 2**

Prerequisite: Students must have knowledge of C programming and working of the computer systems.

Objective:

1. To execute shell scripts in UNIX based operating system.
2. To implement inter process communication using system calls.
3. To implement algorithms for CPU scheduling as well as process synchronization learn and be able to implement the front-end and back-end web-technologies.

Learning Outcomes:

Course Outcome	**Description**
CO1	Understand and execute basic commands of shell script.
CO2	Apply basic operations in shell scripts which are required for different applications.
CO3	Identify and understand concept of file systems in shell script.
CO4	Apply concept of creating new process from parent process.
CO5	Apply concept of virtual file and execute basic commands on it.
CO6	Design communication mechanisms ipc and pipe on linux.

Course Content:

Unit-1; Comparative Study of different operating systems

Unit-2: Demonstration of multitasking concept.

Unit-3: Implementing various process creation algorithms(FCFS,SJF and Round-Robin

Scheduling)

Unit-4: Implementation of memory allocation policies.

Unit-5: Implementing Page replacement algorithms (FIFO,LIFO)

Unit-6: Implementing segmentation algorithms

Unit-7: Implementing file-handling algorithms

Unit-8: Demonstration of working of distributed OS environment.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-4**
P-2		15 Marks	**Based on Lab Exercises: 5-8**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Study material of Web Technology Lab (will be added time to time): Digital copy will be available on the JUET server.

Text Book:

1. "Operating System Concepts"; A. Silberschatz , P. B. Galvin & G. Gagne , Wiley 10e 2018.
2. "Operating Systems: Internals and Design Principles"; W. Stallings, Pearson 9e, 2017.

Reference Books/Material:

1. "Real time systems design and analysis"; P. A. Laplante & S. J. Ovaska, Wiley, 2013.
2. "Real time systems: Theory and Practice"; Mall R., Pearson, 2e, 2009.

Title of Course: AI and Application Lab **Course Code: 18B17CI475**
L-T-P Scheme: **0-0-2** **Credits:** 1

Objectives: In this course we will practical knowledge of the basic components of an intelligence system, their functions, mechanisms, policies and techniques used in their implementation and examples.

Learning Outcomes:

Course Outcome	**Description**
CO1	Apply various AI search algorithms (uninformed, informed, heuristic, constraint satisfaction,).
CO2	Understand the fundamentals of knowledge representation, inference and theorem proving using AI tools.
CO3	Demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information.
CO4	Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems

Course Content:

- Introduction to Python. [Quick Introduction of List, Tuple, Dictionary etc.]

1. Problem solving Agents using Python
    - Simple Reflex Agent
    - Table Driven Agent
    - Random Agent.
    - Goal-Based Agent
    - Utility Based Agent

2. Problem Spaces and blind search techniques by using Python.
    - Breadth First Search
    - Depth First Search
    - Uniform Cost Search
    - Depth Limit Search
    - Recursive Depth Limit Search
    - Iterative Deepening Search

3. Informed search techniques by using Python.
    - Greedy Best First Search
    - A* Search

4. Beyond Classical Search

    - Hill-Climbing Search Algorithm

5. Game playing by using Python.
6. Constraint satisfaction problems by using strawberry prolog.

7. Logic programming by using Python.

Teaching Methodology:

Project Application based lectures would be interactive, and it would cover the core concepts that are explained in the text and reference materials with adequate examples. Demo of tree graph search tutorials will have conceptual and numerical questions that would aid in strengthening the Artificial system principles.

Evaluation Scheme:

Component & Nature	Duration	Marks / Weightage
Practical Test – 1	2 hrs	15
Practical Test – 2	2 hrs	15
Lab Performance	---	10
Day to Day work	---	45
Attendance & Punctuality	----	15
Total		100

Text Books:

1. Allen B. Downey, Think Python, O'Reilly Media
2. Rich, Elaine Knight, Kevin, Artificial Intelligence, Tata McGraw Hill.
3. Luger, George F, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education.

Reference Books:

1. Russell, Stuart J. Norvig, Peter, Artificial Intelligence: A Modern Approach, Pearson Education

Title of Course: Computer Networks Lab **Course Code: 18B17CI471**
L-T-P Scheme: 0-0-2 **Credits:** 1

Course Objectives

- To understand the working principle of various communication protocols.
- To analyze the various routing algorithms.
- To know the concept of data transfer between nodes.

Learning Outcomes:

Course Outcome	Description
CO1	Understand fundamental underlying principles of computer networking
CO2	Understand details and functionality of layered network architecture.
CO3	Apply mathematical foundations to solve computational problems in computer networking
CO4	Analyze performance of various communication protocols.
CO5	Compare routing algorithms
CO6	Practice packet /file transmission between nodes.

Course Content:

1. Identification of network hardware.
2. Fabrication of network cables and trouble shooting.
3. To study *stop &wait* and *sliding window* protocol.
4. To study MAC ALOHA protocol.
5. To study MAC CSMA and MAC CSMA/CD protocol.
6. To study TOKEN BUS and TOKEN RING.
7. To study ETHERNET.
8. To study TOKEN RING.
9. To study SWITCHED LAN.
10. To study Static routing.
11. To study dynamic routing.

Text Books:

1. Sybex CCNA Cisco Certified Network Associate Study Guide.5th Edition
2. Forouzen, Behrouz A.Fegan, Sophia Chung Data Communications and Networking TMH

Reference Books:

1. Carne, E. Bryan Professional's Guide to Data Communication in a CP/IP World Artech House, London, 2004

Title of Course: Mobile and Application Development Lab **Course Code:** 18B28CI408
L-T-P scheme: 1-0-2 **Credit: 3**

Prerequisite: Students must have already registered for the course, "***Introduction to Computers and Programming***" and "**Object Oriented Programming**".

Objective:

1. To learn and be able to implement different mobile-technologies.
2. To develop the abilities to call oneself mobile application developer.

Learning Outcomes:

At the end of the course, a student will:

1. Get familiar with different approaches to mobile application development.
2. Get to learn about application marketing.
3. Have a good grounding of mobile application development requirements, models and IDEs.
4. Possess demonstrative skills in building native applications.
5. Be able to design and develop cross-platform applications.
6. Learn to work in a team on a project.

Course Content:

Part-1: Orientation and Fundamentals of Development

Unit-1 Mobile applications and different approaches to mobile application development. Java features and review of Object Oriented Programming fundamentals.

Part-2: Android Studio and Basic Development Skills

Unit-2 Installing and getting accustomed to the android studio environment. Using activities and views. Working on different views like TextViews, ImageViews etc. Creating simple applications using basic view types.

Unit-3 Using animations, audio and video. Advanced android features like list views, Exception handling, Timers in androids, Advanced String manipulations.

Part-3: Serious Development

Unit-4 Maps and GeoLocation, Storing data permanently, Alert dialogs, SQLite databases, Advanced SQLite, Webviews.

Unit-5 Submitting app to distribution channels, marketing mobile app, Mobile App development models.

Part-4: Working in a team and Cross Platform Development

Unit-6 Using Git, Common Git commands, Project Development, Cross Platform Development using Flutter, Coding using Dart, MVC design pattern, Networking, Data storage, Authentication, State Management.

Teaching Methodology:

This course is introduced to help students transition from a regular developer to a mobile app developer. Starting from the basics, the student will slowly progress to become to other aspects of development including database, version control and other essential technologies that are helpful for a developer. The entire course is broken down into four separate parts: Orientation and Fundamentals of Development, Android Studio and Basic Development Skills, Serious Development, and Working in a team and Cross Platform Development. Each section includes multiple technologies to help a student gain more experience as a developer. This lab course is well complemented by a lecture in the same semester that helps a student learn and discuss the technical details of the underlying technologies.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-7**
P-2		15 Marks	**Based on Lab Exercises: 8-14**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Mobile Development (will be added from time to time): Digital copy will be available on the JUET server.

Books:

Text Book

i. Hello, Android (3rd edition): Introducing Google's Mobile Development Platform by Ed Burnette ISBN: 978-1-93435-656-2

ii. Android Programming for Beginners: Build in-depth, full-featured Android 9 Pie apps starting from zero programming experience, 2nd Edition by John Horton ISBN: 978-1789538502

iii. Head First Android Development: A Brain-Friendly Guide 1st Edition by Dawn Griffiths, David Griffiths. ISBN: 978-1449362188

Reference Books

1. Android Programming: The Big Nerd Ranch Guide (3rd Edition) (Big Nerd Ranch Guides) 3rd Edition by Bill Phillips, Chris Stewart, Kristin Marsicano ISBN: 978-0134706054
2. The Busy Coder's Guide to Android Development Version 8.0 by Mark M Murphy (O nline Book)

Web References:

1. https://developer.android.com
2. https://www.androidauthority.com
3. https://www.vogella.com

Journals:

1. International Journal of Interactive Mobile Technologies (iJIM)
2. ACM Transactions on the Information Systems (TOIC).
3. International Journal of Modern Computer Science (IJMCS)
4. ACM Transactions on Internet Technology (TOIT).