



Maharashtra Education Society's

Abasaheb Garware College, Pune.

(Autonomous)

(Affiliated to Savitribai Phule Pune University)

**Fourth Years B.Sc. Computer Science
(Faculty of Science and Technology)**

Syllabi under Autonomy

B.Sc. (CS) – IV year

Choice Based Credit System Syllabus

To be implemented from Academic Year 2026-2027

Structure of the Course: B. Sc. (Computer Science) IV year

Year	Semester	Course Type	Course Code	Course Title	Remark	Credit	No. of hours to be conducted
4	VII	Core	CS-401-MJ	Paradigms of Programming Languages		4	48
			CS-402-MJ	Design And Analysis of Algorithms		4	48
			CS-403-MJ	Advanced Databases		4	48
			CS-404-MJP	Lab course based on CS-401-MJ and CS-403-MJ		2	4 per batch
		Elective	CS-410-MJ	Cloud Computing		2	30
			CS-411-MJP	Lab Course Based on Cloud Computing		2	4 per batch
			OR				
			CS-412-MJ	Artificial Intelligence		2	30
			CS-413-MJP	Lab Course Based on Artificial Intelligence		2	4 per batch
			OR				
			CS-414-MJ	Web Services		2	30
			CS-415-MJP	Lab Course Based on Web Services		2	4 per batch
			CS-440-RM	Research Methodology		4	48
	VIII	Core	CS-451-MJ	Advanced Operating Systems		4	48
			CS-452-MJ	Mobile Technology		4	48

			CS-453-MJ	Full Stack Development - I		4	48
			CS-454-MJP	Lab course based on CS-552-MJ and CS-553-MJ		2	4 per batch
		Elective	CS-460-MJ	Human Computer Interaction		2	30
			CS-461-MJP	Lab Course Based on Human Computer Interaction		2	4 per batch
			OR				
			CS-462-MJ	Soft Computing		2	30
			CS-463-MJP	Lab Course Based on Soft Computing		2	4 per batch
			CS-481-OJT	On Job Training		4	-

FOURTH YEAR - SEMESTER-VII
CS-401-MJ : Paradigms of Programming Languages

Lectures: 48 (Credits-4)

Prerequisites:

- Procedural Language like C
- Object-Oriented Languages (C++ and Java)
- Concepts of Operating Systems
- Basic Data Structures and Algorithms

Course Objectives:

1. Separate syntax from semantics
2. Compare programming language designs
3. Understand their strengths and weaknesses
4. Learn new languages more quickly
5. Understand basic language implementation techniques

Learning Outcomes:

After completion of the course students will be able to

1. understand different programming language concepts
2. write efficient programs using different programming languages

Unit 1: Introduction

02

- 1.1 The Art of Language Design
- 1.2 The Programming Language Spectrum
- 1.3 Why Study Programming Languages?
- 1.4 Compilation and Interpretation
- 1.5 Programming Environments

Unit 2: Names, Scopes and Bindings

05

- 2.1 The Notion of Binding Time
- 2.2 Object Lifetime and Storage Management
- 2.3 Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection Scope Rules
- 2.4 Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping, The meaning of Names in a Scope
- 2.5 Aliases, Overloading, Polymorphism and Related Concepts, Binding of Referencing Environments
- 2.6 Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures, Macro Expansion

Unit 3: Control Flow

05

- 3.1 Expression Evaluation, Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation
- 3.2 Structured and Unstructured Flow, Structured Alternatives to Go To
- 3.3 Sequencing
- 3.4 Selection - Short-Circuited Conditions, Case/Switch Statements Iteration
- 3.5 Iteration - Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion
- 3.6 Recursion - Iteration and Recursion, Applicative- and Normal-Order Evaluation

Unit 4: Data Types

08

- 4.1. Introduction
- 4.2. Primitive Data Types
- 4.3. Numeric Types: Integer, Floating point, Complex, Decimal, Boolean Types, Character Types
- 4.4. Character String Types
- 4.5. Design Issues, Strings and Their Operations, String Length Operations, Evaluation, Implementation of Character String Types
- 4.6. User defined Ordinal types Enumeration types, Designs Evaluation Subrange types, Ada's design Evaluation Implementation of user defined ordinal types
- 4.7. Array types
- 4.8. Design issues, Arrays and indices, Subscript bindings and array categories, Heterogeneous arrays, Array initialization, Array operations, Rectangular and Jagged arrays, Slices, Evaluation, Implementation of Array Types
- 4.9. Associative Arrays
- 4.10. Structure and operations, Implementing associative arrays,
- 4.11. Record types
- 4.12. Definitions of records, References to record fields, Operations on records, Evaluation, Implementation of Record types
- 4.13. Union Types
- 4.14. Design issues, Discriminated versus Free unions, Evaluation, Implementation of Union types
- 4.15. Pointer and Reference Types
- 4.16. Design issues, Pointer operations, Pointer problems, Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation
- 4.17. Implementation of pointer and reference types - Representation of pointers and references Solution to dangling pointer problem Heap management

Unit 5: Subprograms and Implementing Subprograms

05

- 5.1 Introduction
- 5.2 Fundamentals of Subprograms
- 5.3 Design Issues for subprograms
- 5.4 Local Referencing Environments
- 5.5 Parameter-Passing Methods
- 5.6 Parameters That Are
- 5.7 Subprograms
- 5.8 Overloaded Subprograms
- 5.9 Generic Subroutines, Generic Functions in C++, Generic Methods in Java
- 5.10 Design Issues for Functions
- 5.11 User-Defined Overloaded Operators
- 5.12 Coroutines
- 5.13 Implementing Subprograms
- 5.14 The General Semantics of Calls and Returns
- 5.15 Implementing "Simple" Subprograms
- 5.16 Implementing Subprograms with Stack-Dynamic Local Variables

5.17	Nested Subprograms	
5.18	Blocks	
5.19	Implementing Dynamic Scoping	
Unit 6: Data Abstraction and Object Orientation		08
6.1	Object-Oriented Programming	
6.2	Encapsulation and Inheritance	
6.3	Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance	
6.4	Initialization and Finalization	
6.5	Choosing a Constructor, References and Values, Execution Order, Garbage Collection	
6.6	Dynamic Method Binding	
6.7	Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures	
6.8	Multiple Inheritance	
6.9	Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance	
Unit 7: Concurrency		05
7.1	Introduction: Multiprocessor Architecture	
7.2	Categories of concurrency, Motivations for	
7.3	studying concurrency	
7.4	Introduction to Subprogram-level, concurrency Fundamental concepts, Language Design for concurrency, Design Issues	
7.5	Semaphores - Introduction Cooperation synchronization, Competition Synchronization, Evaluation	
7.6	Monitors - Introduction, Cooperation synchronization, Competition Synchronization, Evaluation,	
7.7	Message Passing Introduction- The concept of Synchronous Message Passing	
7.8	Java Threads - The Thread class -Priorities, Competition Synchronization Cooperation Synchronization, Evaluation	
Unit 8: Functional Programming in Scala		10
8.1	Strings	
8.2	Numbers	
8.3	Control Structures	
8.4	Classes and Properties	
8.5	Methods	
8.6	Objects	
8.7	Functional Programming	
8.8	List, Array, Map, Set	

Reference Books:

1. Programming Language Pragmatics, 3e by Michel L. Scott, Publication: Kaufmann Publishers, An Imprint of Elsevier, USA
2. Concepts of Programming Languages, Eighth Edition by Robert W. Sebesta, Publication: Pearson Education
3. Scala Cookbook by Alvin Alexander, Publication: O'REILLY publication

CS-402-MJ : Design and Analysis of Algorithm

Lectures: 48 (Credits-4)

Prerequisites:

- Basic knowledge of algorithms, programming concepts and Data Structures

Course Objectives:

1. To learn basic Algorithm Analysis techniques and understand the use of asymptotic notation
2. To select the appropriate algorithm by doing necessary analysis of algorithms
3. Understand different algorithm design strategies
4. Understand the use of data structures in improving algorithm performance
5. Understand classification of problems

Learning Outcomes:

After completion of the course students will be able to

1. compare functions using asymptotic analysis and describe the relative performance
2. solve recurrences using the master and the substitution method
3. use the design techniques introduced i.e., dynamic programming, greedy algorithm etc. to design algorithms for more complex problems and analyse their performance.
4. be familiar with the major graph algorithms and their analyses

Unit 1: Basics of Algorithms 04

- 1.1. Algorithm definition and characteristics
- 1.2. Space complexity analysis
- 1.3. Time complexity analysis
- 1.4. Solving recurrence relations using Master and Substitution method

Unit 2: Divide and Conquer Strategy 06

- 2.1 Control abstraction
- 2.2 Binary search with time complexity
- 2.3 Merge sort, Quick sort with best, average and worst-case time complexity analysis
- 2.4 Booth's multiplication algorithm
- 2.5 Strassen's Matrix Multiplication with time complexity

Unit 3: Greedy Method 08

- 3.1 Control abstraction
- 3.2 Fractional Knapsack problem
- 3.3 Job sequencing with deadlines
- 3.4 Minimum-cost spanning trees: Kruskal's and Prim's algorithm
- 3.5 Optimal storage on tape
- 3.6 Optimal merge patterns
- 3.7 Huffman coding
- 3.8 Shortest Path: Dijkstra's Algorithm

Unit 4: Dynamic Programming 12

- 4.1 Principle of optimality
- 4.2 Matrix chain multiplication
- 4.3 0/1 Knapsack Problem
- 4.4 Merge & Purge
- 4.5 Functional Method

4.6	All pairs Shortest Path (Floyd Warshall Algorithm)	
4.7	Bellman ford algorithm	
4.8	Longest common subsequence	
4.9	String editing	
4.10	Boyer moore algorithm	
4.11	Travelling Salesperson problem	
Unit 5: Decrease and Conquer		06
5.1	Graph definition and its representation	
5.2	Graph traversals- DFS and BFS	
5.3	Types of edges	
5.4	Topological sort/order and its application	
5.5	Connected components and strongly connected components	
5.6	Articulation Point and Bridge	
Unit 6: Backtracking		05
6.1	General method	
6.2	Fixed Tuple vs. Variable Tuple formulation	
6.3	n-Queen's problem (n=4)	
6.4	Graph coloring problem	
6.5	Hamiltonian cycle	
6.6	Sum of subsets	
Unit 7: Branch and Bound		05
7.1	Introduction	
7.2	Types of searching (FIFO BB, LIFO, LCBB)	
7.3	Bounding Function, Ranking Function	
7.4	Traveling Salesman problem Using Variable tuple formulation	
7.5	0/1 knapsack problem	
Unit 8: Problem Classification		02
8.1	Deterministic v/s non-deterministic algorithm	
8.2	The class of P, NP, NP-hard and NP - Complete problems	
8.3	Introduction to Approximation Algorithms	

Reference Books:

1. Computer algorithms by Ellis Horowitz, Sartaj Sahani, Sanguthevar Rajasekaran, Publication: Galgotia Publication
2. Algorithms by T. Cormen, C. Leiserson, & R. Rivest, Publication: MIT Press
3. The Design and Analysis of Computer Algorithms by A. Aho, J. Hopcroft & J. Ullman, Publication: Addison Wesley

CS-403-MJ : PSCS-113 Advanced Databases

Lectures: 48 (Credits-4)

Prerequisites:

- Knowledge of file system concepts
- Strong foundation of Related database Concepts (Basic& Advanced)
- A firm foundation of any RDBMS package

Course Objectives:

1. Provide an overview of the concept of NoSQL technology.
2. Provide an insight to the different types of NoSQL databases
3. Make the student capable of making a choice of what database technologies to use, based on their application needs.

Learning Outcomes:

After completion of the course students will be able to -

1. define, compare and use the four types of NoSQL Databases (Document-oriented, Key-Value Pairs, Column-oriented and Graph).
2. explain the detailed architecture, define objects, load data, query data and performance tune Document-oriented NoSQL databases.
3. explain the detailed architecture, define objects, load data, query data and performance tune Graph NoSQL databases.
4. evaluate NoSQL database development tools and programming languages.
5. perform hands-on NoSQL database lab assignments that will allow students to use NoSQL database types via products such as MongoDB and Neo4J.

Unit 1: Introduction to NoSQL

04

- 1.1 Brief History of NoSQL Databases
- 1.2 NoSQL Database Features
- 1.3 Difference between RDBMS and NoSQL
- 1.4 Why NoSQL?
- 1.5 Types of NoSQL Database
 - 1.5.1 Key Value Databases
 - 1.5.2 Document Databases
 - 1.5.3 Column Family Databases
 - 1.5.4 Graph Databases
- 1.6 When should NoSQL be Used?
- 1.7 NoSQL Database Misconceptions

Unit 2: Aggregate Data Models

06

- 2.1 What is an Aggregate?
- 2.2 Differentiate between an Aggregate Data Model and Relational Data Model using an example.
- 2.3 Introduction to Aggregates oriented databases
 - 2.3.1 Key Value Database
 - 2.3.2 Document Database

2.3.3	Column Family Database	
2.4	Consequences of Aggregate Orientation	
Unit 3: Distribution Models		03
3.1	Single Server	
3.2	Sharding	
3.3	Master – Slave Replication	
3.4	Peer to Peer Replication	
3.5	Combining sharding and replication	
Unit 4: CAP Theorem		04
4.1	What are Distributed Database Systems?	
4.2	What is CAP Theorem?	
4.3	Understanding the terms of CAP Theorem	
4.3.1	Consistency	
4.3.2	Availability	
4.3.3	Partition Tolerance	
4.4	Understanding CAP Theorem with an example	
Unit 5: Map Reduce		04
5.1	Basic Map-Reduce	
5.2	Partitioning and Combining	
5.3	Composing Map-Reduce Calculations	
5.4	Two stage Map-Reduce	
5.5	Incremental Map-Reduce	
Unit 6: Advanced Features of NoSQL		06
6.1	Schema Migration	
6.2	Version Stamps	
6.3	Polyglot Persistence	
6.4	Databases on Cloud	
6.5	Database Optimization	
6.6	Choosing your database	
Unit 7: Document Database – MongoDB		11
7.1.	MongoDB Overview	
7.2.	Advantages of MongoDB	
7.3.	Environment Setup	
7.4.	Data Modelling in MongoDB	
7.5.	Database and Collection Creation/Update/Deletion	
7.6.	Data Types	
7.7.	Insert / Update / Delete Document	
7.8.	Projection / Limit / Sort Records in MongoDB	
7.9.	Relationships / Covered Queries / Indexing / Aggregation / Replication /	
7.10.	Object ID / Map Reduce / Regular Expressions	
7.11.	Capped Collection / Auto Increment Sequence	
7.12.	Connecting to MongoDB using Java.	
7.13.	Lots of Practice Query Examples	
7.14.	A Simple Case Study	
Unit 8: Graph Database – Neo4J		10
8.1	Overview	
8.2	Data Model	

- 8.3 Environment Setup
- 8.4 Building Blocks
- 8.5 Introduction to CQL (Cypher Query Language)
- 8.6 Creating nodes and relationships using CQL
- 8.7 Merge / Set / Delete / Remove / Match / For Each / Where / Count
- 8.8 Return / Order by / Limit / Skip / String Functions / Aggregation
- 8.9 Lots of Practice Query Examples
- 8.10 A Simple Case Study

Reference Books:

1. NoSQL Distilled by Pramod Sadalge, Martin Fowler, Publication: Pearson Education
2. NoSQL for Dummies - A Willy Brand
3. MongoDB Cookbook Second Edition by Cyrus Dasadia, Amol Nayak, PCKT Publication
4. Beginning Neo4j by Chris Kemper, Publication: Apress

CS-404-MJP : Lab Course Based on CS-401-MJ and CS-403-MJ

(Credits- 04)

List of PPL Assignments:

1. Assignments of Control Structures
2. Assignments on Arrays
3. Assignments on String
4. Assignments on Classes and Objects
5. Assignments of Lists
6. Assignments on Maps
7. Assignments on Sets

List of Advanced Databases (MongoDB) Assignments:

1. Assignment on Movie Database
Creating Collections Film and Actor and inserting documents in both the collections.
2. Assignment on Company Database
Creating Collections Employee and Transaction and inserting documents in both the collections
3. Write Queries on Movie Database
4. Write Queries on Company Database

List of Advanced Databases (Neo4J) Assignments:

1. Create a library database as graph model.
2. Create a song database as graph model
3. Create an employee database as graph model
4. Create a social network database as graph model
5. Write simple queries on all graph models
6. Write Complex queries on graph models

CS-410-MJ : Cloud Computing

Lectures: 30 (Credits-2)

Prerequisites:

- Operating System
- Fundamentals of Computer Networks
- Good Understanding of Object-Oriented Programming Concepts

Course Objectives:

1. To learn the principles and paradigm of Cloud Computing
2. To appreciate the role of Virtualization Technologies
3. To develop ability to design and deploy Cloud Infrastructure

Learning Outcomes:

After completion of this course student will able to –

1. understand core concepts of cloud computing paradigm.
2. get into system, network and storage virtualization and outline their role in enabling the cloud computing system mode.
3. apply fundamental concepts in cloud infrastructures to understand the trade-offs in power, efficiency and cost.

Unit 1: Introduction to Cloud Computing

08

- 1.1 Overview
- 1.2 Layers and Types of Cloud
- 1.3 Desired Features of a Cloud
- 1.4 Benefits and Disadvantages of Cloud Computing
- 1.5 Cloud Infrastructure Management
- 1.6 Infrastructure as a Service Providers
- 1.7 Platform as a Service Providers
- 1.8 Multitenant Technology.
- 1.9 Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Centre Technology, Virtualization Technology
- 1.10 Infrastructure as a Service, Platform as a Service, Software as a Service, Cloud Deployment Models.

Unit 2: Abstraction and Virtualization

07

- 2.1 Introduction to Virtualization Technologies
- 2.2 Load Balancing and Virtualization
- 2.3 Understanding Hypervisors
- 2.4 Virtual Machines Provisioning and Manageability Virtual Machine Migration Services
- 2.5 Provisioning in the Cloud Context Virtualization of CPU, Memory, I/O Devices, Virtual Clusters and Resource management

Unit 3: Programming, Environments and Applications

08

- 3.1 Features of Cloud and Grid platforms
- 3.2 Programming Support of Google App Engine
- 3.3 Programming on Amazon AWS and Microsoft Azure
- 3.4 Emerging Cloud Software Environments
- 3.5 Applications: Moving application to cloud
- 3.6 Microsoft Cloud Services
- 3.7 Google Cloud Applications
- 3.8 Amazon Cloud Services
- 3.9 CI/CD pipelines for microservices
- 3.10 Cloud Applications.

Unit 4: Security in The Cloud

07

- 4.1 Security Overview – Cloud Security
- 4.2 Challenges and Risks
 - 4.2.1 Software-as-a-Service Security
 - 4.2.2 Security Governance
 - 4.2.3 Risk Management
 - 4.2.4 Security Monitoring
 - 4.2.5 Security Architecture Design
 - 4.2.6 Data Security
 - 4.2.7 Application Security
 - 4.2.8 Virtual Machine Security
- 4.3 Identity Management and Access Control
- 4.4 Disaster Recovery in Clouds

Reference Books:

1. Cloud Computing: Technologies and Strategies of the Ubiquitous Data Centre by Brian J.S. Chee and Curtis Franklin
Publication: CRC Press, ISBN :9781439806128
2. Mastering Cloud Computing: Foundations and Applications Programming by Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi
Publication: McGraw Hill, ISBN: 978 1259029950, 1259029956
3. Distributed and Cloud Computing, From Parallel Processing to the Internet of Things by Kai Hwang, Geoffrey C Fox, Jack G Dongarra
Publication: Morgan Kaufmann Publishers, 2012.

CS-411-MJP : Lab Course Based on Cloud Computing

(Credits-2)

List of Sample Assignments:

1. Working and Implementation of Infrastructure as a service.
2. Working and Implementation of Software as a service.
3. Working and Implementation of Platform as a service.
4. Practical Implementation of Storage as a Service.
5. Working of Google drive to make spreadsheet and notes.
6. Working and Implementation of identity management.
7. Write a program for web feed.

8. Execute the step to Demonstrate and implementation of cloud on single sign on.
9. Practical Implementation of cloud security.
10. Installing and Developing Application Using Google App Engine.
11. Implement VMWareESXi Server
12. Using OpenNebula to manage heterogeneous distributed data center Infrastructure.
13. Implementation of Cloud Failure Cluster.
14. Managing and working of cloud xen server.
15. Working with Aneka and demonstrate how to Managing cloud computing Resources.
16. Installation and configuration of cloud Hadoop and demonstrate simple query.
17. Create a sample mobile application using Amazon Web Service (AWS) account as a cloud service. Also provide database connectivity with implemented mobile application.

CS-412-MJ : Artificial Intelligence

Lectures: 30 (Credits-2)

Prerequisites:

- Basic concepts of algorithms and Data Structures.
- Basics of Formal Logic.
- Any One Programming C/CPP/ Java/ Python.

Course Objectives:

1. To understand a historical perspective of AI and its foundations.
2. To learn principles of AI.
3. To learn of different strategies and implementing them to solve problems.

Learning Outcomes:

After completion of this course students will able to -

1. solve problems using AI techniques.
2. ready with foundation required to develop Expert Systems.

Unit 1: Introduction to Artificial Intelligence	04
1.1 The History of Artificial Intelligence	
1.2 Applications of Artificial Intelligence	
1.3 Intelligent Agents, Types of Agents	
1.4 Concept of State Space Search	
1.5 Control Strategies	
1.6 Problem Characteristics	
1.7 Issues in Design of Search Program	
1.8 Production System and Water-Jug Problem	
Unit 2: Uninformed Search Techniques	04
2.1 DFS, BFS, Depth Limited Search	
2.2 Uniform Cost Search.	
2.3 Iterative Deepening	
2.4 Bidirectional Search	
Unit 3: Informed Search Techniques	06
3.1 Introduction to Heuristic	
3.2 Hill Climbing	
3.3 Best First Search	
3.4 A*, AO*, IDA*, SMA*	
Unit 4: Adversarial Search	04
4.1 Game Theory	
4.2 Two-player zero-sum games	
4.3 Min-Max Search	
4.4 Alpha Beta Pruning	
4.5 Limitations of Game Search Algorithms	

Unit 5: Knowledge Representation (KR) and Reasoning	10
5.1 KR in Propositional Logic	
5.2 KR in Predicate Logic	
5.3 Representing Simple facts in Logic	
5.4 Resolution	
5.5 Forward and backward chaining	
5.6 Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM)	
5.7 Bayesian Networks	
5.8 Belief Networks, Simple Inference in Belief Networks	
Unit 6: Expert System	02
6.1 Concept	
6.2 Expert System Shells	
6.3 Explanation	
6.4 Knowledge Acquisition	

Reference Books:

1. Artificial Intelligence, 2nd Edition by Elain Rich and Kevin Knight
Publication: Tata McGraw Hill, 1995
2. Artificial Intelligence a Modern Approach, 2nd Edition by Stuart Russel and Peter Norvig
Publication: Pearson Education, 2003 / PHI
3. A Guide to Expert Systems by Donald A. Waterman
Publication: Pearson Education

CS-413-MJP : Lab Course Based on Artificial Intelligence
(Credits-2)

List of Sample Assignments:

(Student can choose any programming language to do assignments.)

1. Write a program to implement Breadth First Search Traversal.
2. Write a program to implement Depth First Search Traversal.
3. Write a program to implement A*
4. Write a program to count total number of goal states
5. Write a program to implement uniform cost search
6. Write a program to implement Water Jug Problem.
7. Implement two player game using minimax search algorithm.
8. Write a program to calculate n from initial state and goal state of 8-puzzle problem.
'n' is the number of tiles that are not at expected place according to goal state.

Case Study - It should be done individually on the topic of Knowledge Representation and reasoning.

CS-414-MJ : Web Services

Lectures: 30 (Credits-2)

Prerequisites:

- Strong knowledge about Java programming.
- Good Understanding of Object- Oriented Programming concepts.
- Must be familiar with XML.

Course Objectives:

1. To learn the details of web services technologies like WSDL, UDDI, SOAP
2. To learn how to implement and deploy web service client and server
3. To explore interoperability between different frameworks
4. To get ready for creating web service using OOP concept

Learning Outcomes:

After completion of this course student will be able to –

1. understand the principles of SOA, SOAP.
2. efficiently use market leading environment tools to create and consume web services.
3. identify and select the appropriate framework components in creation of webservice solution
4. apply OOP principles to creation of web service solutions.

Unit 1: Web Service and SOA fundamentals Introduction to Web Services 06

- 1.1 The definition of web services
- 1.2 Basic operational model of web services
- 1.3 Tools and technologies enabling web services
- 1.4 Benefits and challenges of using web services.
- 1.5 Web Services Architecture
 - 1.5.1 Web services Architecture and its characteristics
 - 1.5.2 Core building blocks of web services
 - 1.5.3 Standards and technologies available for implementing web services
 - 1.5.4 Web services communication models
 - 1.5.5 Basic steps of implementing web services.

Unit 2: SOAP: Simple Object Access Protocol 06

- 2.1 Inter-application communication and wire protocols
- 2.2 SOAP as a messaging protocol
- 2.3 Structure of a SOAP message
- 2.4 SOAP communication model
- 2.5 Building SOAP Web Services
- 2.6 Developing SOAP Web Services using Java
- 2.7 Error handling in SOAP
- 2.8 Advantages and disadvantages of SOAP.

Unit 3: Describing and Discovering Web Services 08

- 3.1 WSDL - WSDL in the world of Web Services
- 3.2 Web Services life cycle
- 3.3 Anatomy of WSDL definition document
- 3.4 WSDL bindings, WSDL Tools, limitations of WSDL
- 3.5 Service discovery, role of service discovery in a SOA
- 3.6 Service discovery mechanisms

- 3.7 UDDI – UDDI Registries, uses of UDDI Registry,
- 3.8 Programming with UDDI
- 3.9 UDDI data structures
- 3.10 Support for categorization in UDDI Registries
- 3.11 Publishing API, Publishing information to a UDDI Registry, searching information in a UDDI Registry, deleting information in a UDDI Registry, limitations of UDDI.

Unit 4: The REST Architectural style

10

- 4.1 Introducing HTTP
- 4.2 Core architectural elements of a RESTful system
- 4.3 Description and discovery of RESTful web services
- 4.4 Java tools and frameworks for building RESTful web services
- 4.5 JSON message format and tools and frameworks around JSON
- 4.6 Build RESTful web services with JAX-RS APIs
- 4.7 The Description and Discovery of RESTful Web Services
- 4.8 Design guidelines for building RESTful web services
- 4.9 Secure RESTful web services
- 4.10 Understanding token-based authorization mechanism and implement using REST API.

Reference Books:

1. Building Web Services with Java, 2nd Edition by S. Graham and others
Publication: Pearson Edn., 2008.
2. J2EE Web Services by Richard Monson-Haefel
Publication: Pearson Education.
3. Java Web Services Programming by R.Mogha,V.V.Preetham
Publication: Wiley India Pvt.Ltd.
4. XML, Web Services, and the Data Revolution by F.P.Coyle
Publication: Pearson Education

CS-415-MJP : Lab Course Based on WebServices

(Credits-2)

List of Sample Assignments:

1. Create 'Dynamic Web Project', which will host your web service functionality to greet the user according to server time and create 'Dynamic Web Project', which will host the client application that will send username and test the web service.
2. Create 'Dynamic Web Project', which will host your web service functionality to convert Celsius to Fahrenheit and create 'Dynamic Web Project', which will host the client application that will send Celsius and test the web service.
3. Create 'Dynamic Web Project', which will host your web service functionality to find the factorial of a given number and create 'Dynamic Web Project', which will host the client application that will send a positive integer number and test the web service.
4. Create 'Dynamic Web Project', which will host your web service functionality to validate email id (use regular expression) and create 'Dynamic Web Project', which will host the client application that will send email id and test the web service.
5. Create 'Dynamic Web Project', which will host your web service functionality to validate username and password (use a database for storing username and password) and create 'Dynamic Web Project', which will host the client application that will send username and password and test the web service.
6. Create 'Dynamic Web Project', which will host your web service functionality to select employee details (use a database for storing emp details (eno, ename, designation, salary)) and create 'Dynamic Web Project', which will host the client application that will send employee name and display the details.
7. Create 'Dynamic Web Project', which will host your web service functionality to select Movie details (Movie(mno, mname, release_year) and Actor(ano, aname), 1: M cardinality) and create 'Dynamic Web Project', which will host the client application that will send actor name and display the details.
8. Create 'Dynamic Web Project', which will host your web service functionality to validate mobile no (use regular expression: should contain only 10 numeric no) and create 'Dynamic Web Project', which will host the client application that will send mobile no and test the web service.
9. Create 'Dynamic Web Project', which will host your web service functionality to convert Rupees to Dollar, Pound, Euro,..... and create 'Dynamic Web Project', which will host the client application that will send an amount in Rupees & type of conversion and tests the web service.

CS-440-MJ: Research Methodology

Lectures: 48 (Credits- 04)

Learning Objectives:

1. Research Methodology course is designed to equip students with the necessary knowledge, skills, and understanding of various research techniques and methodologies.
2. Students should be familiar with various data collection techniques, such as surveys, interviews, observations, and experiments, and understand their strengths and limitations.
3. Students should be aware of ethical considerations in research, including issues related to participant consent, privacy, confidentiality, and avoiding plagiarism.
4. Its aim is to enable students to conduct research effectively, critically evaluate existing research, and contribute to the advancement of knowledge in their respective fields.

Course Outcomes: On completion of this course, students will be able to

- C01: Understand of the fundamental concepts of research, including the research process, research questions, hypotheses, and variables.
- C02: Conduct a comprehensive literature review to identify relevant studies, synthesize existing knowledge, and identify research gaps.
- C03: Identify research problems, formulate research questions, and design appropriate methodologies to address these problems
- C04: Identify and select appropriate research designs, such as experimental, observational, survey, qualitative, or mixed-methods, based on the research objectives.
- C05: Apply appropriate data analysis methods, including statistical techniques or qualitative analysis, to draw meaningful conclusions from research data.
- C06: Develop a well-structured research proposal, outlining research questions, methodology, expected outcomes, and a rationale for the study.
- C07: Communicate research findings effectively through written reports, presentations, and academic papers.
- C08: Gain an appreciation for the importance of research in contributing to the advancement of knowledge in their field of study and broader society.
- C09: Understand the principles of research ethics and integrity and apply them in their research.

Unit 1: Introduction to Research Methodology

08 (C01, C09)

- 1.1 Meaning of Research
- 1.2 Objectives of Research
- 1.3 Types of Research
- 1.4 Research Approaches
- 1.5 Significance of Research
- 1.6 Research Ethics and Integrity
- 1.7 Plagiarism and types of plagiarism
- 1.8 Research Methods versus Methodology
- 1.9 Criteria of Good Research

**Unit 2: Literature Review and Formulation of Research Problems
C03)**

06 (C01, C02,

- 2.1 Research Process
- 2.2 Reviewing the literature: purpose of a literature review
- 2.3 Literature resources
- 2.4 The Internet and a literature review
- 2.1 Conducting and Evaluating literature reviews
- 2.2 Formulation of research problem
 - 2.2.1 What is a Research Problem?
 - 2.2.2 Selecting the Problem
 - 2.2.3 Necessity of Defining the Problem
 - 2.2.4 Technique Involved in Defining a Problem

Unit 3: Research Design

06 (C03, C04)

- 3.1 Meaning of Research Design
- 3.2 Need for Research Design
- 3.3 Features of a Good Design
- 3.4 Important Concepts Relating to Research Design
- 3.5 Different Research Designs/Methods
 - 3.5.1 Pure and Applied Research
 - 3.5.2 Exploratory or Formulative Research
 - 3.5.3 Descriptive Research
 - 3.5.4 Diagnostic Research
 - 3.5.5 Evaluation Studies
 - 3.5.6 Action Research
 - 3.5.7 Experimental Research
 - 3.5.8 Analytical Study or Statistical Method
 - 3.5.9 Historical Research
 - 3.5.10 Surveys
 - 3.5.11 Case Study
 - 3.5.12 Field Studies

Unit 4: Hypothesis and Sampling

06 (C05, C06)

- 4.1 What is Hypothesis?
- 4.2 Nature & Characteristics of Hypothesis
- 4.3 Significance of Hypothesis
- 4.4 Types of Hypothesis
- 4.5 Sources of Hypothesis
- 4.6 Characteristics of Good Hypothesis
- 4.7 What is sampling?
- 4.8 Characteristics of Good Sample
- 4.9 Basis of Sampling
- 4.10 Merits and demerits of Sampling
- 4.11 Sampling Techniques or Methods
- 4.12 Sample Design and Choice of Sampling Technique

Unit 5: Data Collection, Processing and Analysis of Data

06 (C05)

- 1.1 Collection of Primary Data
- 1.2 Method of data Collections - Observation, Interview, Questionnaires and Schedules
- 1.3 Difference between Questionnaires and Schedules
- 1.4 Some Other Methods of Data Collection
- 1.5 Collection of Secondary Data
- 1.6 Selection of Appropriate Method for Data Collection
- 1.7 Case Study Method
- 1.8 Processing Operations and Some Problems in Processing
- 1.9 Elements/Types of Data Analysis
- 1.10 Measures of Central Tendency, Dispersion, Asymmetry (Skewness)
- 1.11 Measures of Relationship - Chi-Square, t-test, ANNOVA(f-test), Z-test
- 1.12 Simple Regression Analysis, and Multiple Correlation and Regression
- 1.13 Quantitative and Qualitative Data Analysis Tools

Unit 6: Interpretation and Report Writing

08 (C06, C07, C08)

- 1.1 Meaning of Interpretation, Why Interpretation?
- 1.2 Technique of Interpretation
- 1.3 Precaution in Interpretation
- 1.4 Significance of Report Writing
- 1.5 Different Steps in Writing Report
- 1.6 Layout of the Research Report
- 1.7 Types of Reports (Research Proposal/Synopsis, Research Paper, and Thesis)
- 1.8 Oral Presentation
- 1.9 Mechanics of Writing a Research Report
- 1.10 Precautions for Writing Research Reports

Unit 7: Publication Ethics and Open Access Publishing

08 (C07, C09)

- 7.1 Publication ethics: definition, introduction and importance
 - 7.2 Best practices/standards setting initiatives and guidelines: COPE, WAME, etc.
 - 7.3 Conflicts of interest
 - 7.4 Publication misconduct: definition, concept, problems that lead to unethical behaviour and vice versa, types
 - 7.5 Violation of publication ethics, authorship and contributor ship
 - 7.6 Identification of publication misconduct, complaints and appeals
 - 7.7 Predatory publishers and journal
 - 7.8 Open access publications and initiatives
 - 7.9 SHERPA/RoMEO online resource to check publisher copyright & self-archiving policies
 - 7.10 Software tool to identify predatory publications developed by SPPU
 - 7.11 Journal finder/ journal suggestion tools viz. JANE, Elsevier Journal Finder, Springer Journal Suggester, etc.
- E-Resources for research: GoogleScholar, ShodhGanaga, ShodhGangotri, SciHub, etc...

Reference Books:

- 1. Researching Information Systems and Computing by Briony J Oates, SAGE SOUTH ASIA EDITION
- 2. Research Methodology: A Step-by-Step Guide for Beginners, Kumar, Pearson Education.

3. Research Methodology Methods and Techniques, Kothari, C. R., Wiley Eastern Ltd.
4. The Research Methods Knowledge Base, by William M. K. Trochim, James P. Donnelly
5. Introducing Research Methodology: A Beginner's Guide to Doing a Research Project, by Uwe Flick
6. A Guide to Research and Publication Ethics by Partha Pratim Ray, New Delhi Publishers
7. RESEARCH & PUBLICATION ETHICS by Wakil Kumar Yadav, NOTION PRESS
8. Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd.

SEMESTER-II
CS-451-MJ : Advanced Operating Systems

Prerequisite:

Lectures: 48 (Credits- 04)

- Working knowledge of C programming.
- Basic Computer Architecture concepts.
- Basic algorithms and data structure concepts

Course Objectives:

1. To learn Advanced Operating Systems Concepts using Unix/Linux.
2. To understand the programming interface to the Unix/Linux system - the system call interface.
3. To grasp the concepts underlying in the design and implementation of Operating Systems.

Learning Outcomes:

After completion of the course, students will be able to:

1. implement advanced operating systems concepts in C.
2. develop efficient system software.
3. easier to understand internals of other operating systems and their functionalities.

Unit 1: Introduction to UNIX/Linux Kernel

02

- 1.1 System Structure
- 1.2 User Perspective
- 1.3 Assumptions about Hardware
- 1.4 Architecture of UNIX Operating System
- 1.5 Introduction to System Concepts

Unit 2: Implementation of File Subsystem

10

- 2.1 Buffer headers, structure of the buffer pool, scenarios for retrieval of a buffer
- 2.2 reading and writing disk blocks, inodes, structure of regular file, Directories, Conversion of a Pathname to an i-node, super block, inode assignment to a new file, allocation of disk blocks
- 2.3 open, read, write, lseek, close, creat, mknod, chdir, chroot, chown, chmod, stat, fstat, pipes, dup, mount, umount, link, unlink

Unit 3: Operations on Files

06

- 3.1 open, creat, file sharing, atomic operations, dup2, sync, fsync, and fdatasync, fcntl, /dev/fd
- 3.2 stat, fstat, lstat, file types, Set-User-ID and Set-Group-ID

3.3 file access permissions, ownership of new files and directories	
3.4 access function, umask function, chmod and fchmod	
3.5 sticky bit	
3.6 chown, fchown, and lchown, file size, file truncation, file systems	
3.7 link, unlink, remove, and rename functions, symbolic links, symlink and readlink functions	
3.8 file times, utime, mkdir and rmdir, reading directories, chdir, fchdir, and getcwd, device special files	
Unit 4: Process Control, Scheduling and time	14
4.1 Process states and transitions, Layout of system memory, the context of a process, saving the context of a process, sleep, wakeup	
4.2 Process creation, signals, process termination, awaiting process termination, invoking other programs, the user id of a process, changing the size of the process, The Shell, System boot and the init process	
4.3 Process Scheduling, system calls for time	
Unit 5: Process Environment and Relationships	06
5.1 Process termination	
5.2 Environment list	
5.3 Memory layout of a C program	
5.4 shared libraries, environment variables	
5.5 setjmp and longjmp, getrlimit and setrlimit	
5.6 process identifiers, fork, vfork, exit, wait and waitpid, waitid, wait3 and wait4	
5.7 Race conditions, exec	
5.8 changing user IDs and group IDs, system function, user identification, process times	
Unit 6: Memory Management	06
6.1 The Process Address Space, Allocating Dynamic Memory, Managing Data Segment, Anonymous Memory Mappings, Advanced Memory Allocation, Debugging Memory Allocations, Stack-Based Allocations, choosing a Memory Allocation Mechanism, Manipulating Memory, Locking Memory, OpportunisticAllocation	
6.2 Swapping, Demand Paging	
Unit 7: Signal Handling	04
7.1 Signal concepts, signal function, unreliable signals	
7.2 interrupted system calls, reentrant functions	
7.3 SIGCLD semantics, reliable-signal technology	
7.4 kill and raise, alarm and pause, signal sets, sigprocmask, sigpending, sigsetjmp and siglongjmp, sigsuspend, sigaction, sigqueue	
7.5 abort, system function revisited, sleep, nanosleep, and clock_nanosleep Functions	

Reference Books

1. The Design of the UNIX Operating System, Maurice J. Bach., PHI
2. Advanced Programming in the UNIX Environment, Richard Stevens, Addison- Wesley
3. Linux System Programming, Robert Love,O'Reilly

CS-452-MJ : Mobile Technology

Lectures: 48 (Credits- 04)

Prerequisite:

- Basic knowledge about programming
- Learning of Java or Kotlin
- Concepts of OOPs
- Knowledge about any database management system
- Understanding of XML

Course Objectives:

1. To learn installation and configuration of android development tools
2. To learn to design and development of user interfaces
3. To be able to apply location-based services and to connect to SQLite database
4. To learn to install .apk file on android device

Learning Outcomes:

After completion of this course students will be able to -

1. install and configure Android application development tools.
2. design and develop user Interfaces for the Android platform.
3. apply Java programming concepts to Android application development.
4. do installation of apk file.

Unit 1: Introduction to Android

02

- 1.1 A little Background about mobile technologies
- 1.2 Overview of Android - An Open Platform for Mobile development Open Handset Alliance
- 1.3 Installation of JDK and Android Studio

Unit 2: Developing for Android: My First Android Application

04

- 2.1 How to setup Android Development Environment.
- 2.2 Android development Framework - Android-SDK
- 2.3 Eclipse Emulators – What is an Emulator / Android AVD?
- 2.4 Creating & setting up custom Android emulator
- 2.5 Android Project Framework
- 2.6 My First Android Application

Unit 3: Android Activities and UI Design

10

- 3.1 Understanding Intent, Activity, Activity Lifecycle and Manifest
- 3.2 Creating Application and new Activities
- 3.3 Expressions and Flow control, Android Manifest Simple UI -Layouts and Layout

3.4	Properties, Fundamental Android UI DesignIntroducing Layouts, Creating new Layouts, Drawable Resources	
3.5	Resolution and density independence (px, dip, dp, sip, sp)	
3.6	XML Introduction to GUI objects viz.	
3.7	Push Button, Text / Labels, Edit Text, Toggle Button, Weight Sum Padding, Layout, Weight	
Unit 4:	Advanced UX / UI Programming	08
4.1	Event driven Programming in Android (Text Edit, Button clicked etc.)	
4.2	Android Activity Lifecycle	
4.3	Understanding the Exception	
4.4	Toast, Menu, Dialog, List and Adapters What is Menu?	
4.5	Custom Vs. System Menus	
4.6	Creating and Using Handset menu Button (Hardware)	
4.7	What are Android Themes? What is Dialog? How to create an Alter Dialog?	
4.8	What is Toast in Android?	
4.9	List & Adapters	
4.10	Manifest.xml File Upd	
4.11	Understanding the role of UX design in Android app development.	
Unit 5:	Multimedia Programming and SQLite Database	12
5.1	Multimedia audio formats - Creating and Playing	
5.2	Multimedia audio formats - Kill / Releasing (Memory Management)	
5.3	How to associate audio in any application	
5.4	How to associate video playback with an event	
5.5	Introducing SQLite	
5.6	SQLite Open Helper and creating a database	
5.7	Opening and closing a database	
5.8	Working with cursors Inserts, updates, and deletes	
Unit 6:	Location Based Services and Google Maps Using Location Based Services	06
6.1	Working with Google Maps	
6.2	Notifications	
6.3	Notification Manager	
6.4	Pending Intent Notifications (Show and Cancel)	
6.5	How to develop your own custom-made Web browser	
6.6	How to use Web-view object in XML	
Unit 7:	Android Development using other Tools	06
7.1	Other ways to Develop Android Applications	
7.2	Graphics / Game development using Adobe CS5.5 Flash	

- 7.3 How to render .apk file from Adobe FlashHow to use LogCat (Verbose, Debug, Info, Warn, Error, Assert)
- 7.4 Use of Perspectives
- 7.5 Installation of .apk How to install .apk into your Android Mobile

Reference Books

1. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides), 2013 by Bill Phillips and Brian Hardy
2. Professional Android 4 Application Development, 2012 by Reto Meier
3. Android Application Development in 24 Hours, Sams Teach Yourself (4th Edition), 2015 by Carmen Delessio and Lauren Darcey.

CS-453-MJ : Full Stack Development - I

Lectures: 48(Credits- 04)

Prerequisite:

- Basics of Scripting Language
- Concept of client and server
- Concept of Web Server

Course Objectives:

1. To understand and learn modern web technologies.
2. To understand server-side programming.
3. To learn developing basic interactive web application.

Learning Outcomes:

After completion of this course students will be able to -

1. write programs that are server-side scripting.
2. know about latest web development tools and techniques in IT industry.
3. get ready for advanced level of Full Stack Development Course.

Unit 1: Introduction

04

- 1.1 How Web works?
- 1.2 What is Full Stack Development?
- 1.3 What is ECMA Script?
- 1.4 Concept of Client-Side Scripting
- 1.5 Concept of Server-Side Scripting
- 1.6 Concept of Framework
- 1.7 Concept of UI/UX.

Unit 2: Basics of JavaScript

10

- 2.1 JavaScript Variables and Operators
- 2.2 JavaScript Arrays and Functions
- 2.3 Importance and need of JavaScript Objects
- 2.4 JavaScript Events
- 2.5 Async Functions
- 2.6 Promises and Callback

Unit 3: Introduction to Node.js and NPM

12

- 3.1 Traditional Web Server Model
- 3.2 Concept of Node.js and Event Loop
- 3.3 Installing Node.js
- 3.4 Components of Node.js Application
- 3.5 Modules, Types of Modules
- 3.6 Concept and examples of export

- 3.7 Create a basic application
- 3.8 What is Node Package Manager (NPM)
- 3.9 Packages in Node.js
- 3.10 Local and global Packages
- 3.11 Dependencies in packages

Unit 4: Express.js **12**

- 4.1 REST, RESTful services
- 4.2 Introduction to Express Frameworks
- 4.3 Routing
- 4.4 Routes, Views and Template Engine
- 4.5 Handling HTTP GET/POST request
- 4.6 Calling Endpoints using Postman
- 4.7 Using middleware and types of middleware
- 4.8 Form Fetching Data
- 4.9 Form Validation

Unit 5: Working with Databases **10**

- 5.1 Connection String
- 5.2 Configuring
- 5.3 Working with Select command
- 5.4 Various database operations
- 5.5 MongoDB
- 5.6 Mongoose ODM
- 5.7 Mongoose Schema
- 5.8 Mongoose Model
- 5.9 Querying with Mongoose
- 5.10 Aggregations in mongodb

Reference Books

1. Mastering Nod.js, Sandro Pasquali, Packt
2. Node.js complete reference guide, Velentin Bojinov, David Herron, Dioge Resende, Packt
3. Smashing Node.js, Java Script Everywhere, Guillermo Rauch, WILEY

CS-454-MJP : Lab Course Based on CS-552-MJ and CS-553-MJ

(Credits- 04)

List of Assignments for Mobile Technology

Assignment 1

1. Setting up of Android Studio and Emulator
2. Hello World Application

Assignment 2

1. Use of Activity and Intents
2. Screen design using UI components

Assignment 3

1. Use of Toasts, Dialogs
2. Use of Lists and Adapters

Assignment 4

1. Use of Multimedia files in Android Application
2. Connection with SQLite Database

Assignment 5

1. Working with Google Maps and Notification
2. Installation of .apk file on Android Device.

List of Assignments for Full Stack Development – I

Assignment 1

JavaScript

Assignment 2

Basics of Node.js

Assignment 3

Based on NPM concept

Assignment 4

Node.js programs with DB

Assignment 5

Based on Express.js

Mini Project can be given either for MT or FSD – I.

CS-460-MJ : Human Computer Interaction

Lectures: 30 (Credits- 02)

Prerequisite:

- Foundations of Human Computer Interaction
- Be familiar with the design technologies for individuals and persons with disabilities
- Be aware of mobile HCI
- Learn the guidelines for user interface.

Course Objectives:

1. To design effective dialog for HCI.
2. To identify the impact of usable interfaces in the acceptance and performance utilization of information systems.
3. To give insight into the research area.

Learning Outcomes:

After completion of this course students will be able to -

1. understand what HCI design is.
2. develop interfaces ranging from WIMPs (windows, icons, menus, pointers) to wearables

Unit 1: Foundations of HCI

06

- 1.1 The Human: I/O channels – Memory – Reasoning and problem solving
- 1.2 The computer: Devices – Memory – processing and networks
- 1.3 Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity-Paradigms.

Unit 2: Design & Software Process

07

- 2.1 Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping
- 2.2 HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale
- 2.3 Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design

Unit 3: Models and Theories

05

- 3.1 Cognitive models
- 3.2 Socio-Organizational issues and stake holder requirements
- 3.3 Communication and collaboration models

- 3.4 Hypertext
- 3.5 Multimedia and www.

Unit 4: Mobile HCI

06

- 4.1 Mobile Ecosystem: Platforms, Application frameworks
- 4.2 Types of Mobile Applications: Widgets, Applications, Games
- 4.3 Mobile Information Architecture
- 4.4 Mobile 2.0
- 4.5 Mobile Design: Elements of Mobile Design, Tools.

Unit 5: Web Interface Design

06

- 5.1 Designing Web Interfaces
- 5.2 Drag & Drop, Direct Selection
- 5.3 Contextual Tools
- 5.4 Overlays
- 5.5 Inlays and Virtual Pages
- 5.6 Process Flow
- 5.7 Case Studies

Reference Books:

1. Human Computer Interaction, (Chapter 1, 2 & 3) Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, 3rd Edition, Pearson Education, 2004
2. Mobile Design and Development, (Chapter 4) Brian Fling, First Edition, O'Reilly Media Inc., 2009
3. Designing Web Interfaces, (Chapter 5) Bill Scott and Theresa Neil, First Edition, O'Reilly, 2009

CS-461-MJP : Lab Course Based on Human Computer Interaction

(Credits- 02)

Note: Any tool or technology can be used for implementation e. g. JAVA, PHP, etc.

Sample Questions –

1. Understand the trouble of interacting with Computers - Redesign interfaces of applications. Select any application, like land-line phone application, registration etc and understand the trouble of interacting with that application. Comment on design of that application as good or bad design based on whether interaction principles are matching with users' mental model or not. Redesign the interface for mention the change in design and reason.
2. Know your client: Select anyone category of user and develop application understanding the user who will be using your system. Comment on the category of user selected and specific features given for the users and identify what kinds of interfaces will they like and why?. Compare with existing system analyze and rate them. Analyze user models and develop user centric interfaces for :
 - a. Children (4-5 years of age): An application to teach math. Perform analysis of children behaviour e.g. their preferences, interests etc.
 - b. Teenagers: Design a digital diary for young teens to help them overcome various social pressures they deal with during their teen years. The diary should also be like a self help tool which would help them deal with incidents like bullying, peer pressure, etc.. This is an open project and you can think in any direction to make the children sail through their teen years while trying to discover life around them. Perform analysis of teenagers e.g. their problems, interests, needs, etc.
 - c. Older generation: Folks from the older generation has been very wary of using their credit card on the Internet. They have various concerns when it comes to paying their bills. Also because of their old age, it will be beneficial for them to use the internet and pay their phone, electricity, gas, etc. bills. Analysis of old people e.g. their nature, interests, needs, etc.
 - d. Rural people: ATVM for train ticketing in rural area. Perform analysis of rural people e.g. their problems, interests, needs, language etc.
 - e. Mentally disabled: Design the interface of a game for mentally disabled children. Analysis of mentally disabled e.g. their behaviour, problems, interests... Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.
3. Identify 5 different websites catering to one specific goal (eg. Goal – on-line shopping and 5 different websites – ebay, amazon, flipkart, zovi, myntra) and perform a competitive analysis on them to understand how each one caters to the goal, the interactions and flow of the payment system and prepare a report on the same. Consider any 8 HCI principles and prepare the following table evaluating the websites.

Sr. No	Principles	Poor	Average	Good	Good Very	Excellent
1.	Aesthetically pleasing					
2.	..					

4. To achieve simplicity one needs to optimize the number of elements on a screen, within limits of clarity. And minimize the alignment points, especially horizontal or columnar
 - a. Calculate Screen Complexity for existing Graphical User Interface (GUI).
 - b. Redesign the Screen by applying various guidelines to lower the complexity of selected Graphical User Interface (GUI) to achieve simplicity

Start by designing on paper, not on the computer. Ask everyone to sketch his or her ideas.

- Draw in black and white: Many icons will be displayed in monochrome. Color is an enhancing property; consider it as such.
- Test for expectation, recognition, and learning. Choosing the objects and actions, and the icons to represent them, is not a precise process, and will not be easy. So, as in any screen design activity, adequate testing and possible refinement of developed images must be built into the design process. Icon recognition and learning should both be measured as part of the normal testing process.
- Test for legibility.
- Verify the legibility and clarity of the icons in general. Also, verify the legibility of the icons on the screen backgrounds chosen. White or gray backgrounds may create difficulties. An icon mapped in color, then displayed on a monochrome screen, may not present itself satisfactorily. Be prepared to redraw it in black and white, if necessary.
- Register new icons in the system's registry.
- Create and maintain a registry of all system icons. Provide a detailed and distinctive description of all new icons.

CS-462-MJ : Soft Computing

Lectures: 30 (Credits- 02)

Prerequisite:

- A strong mathematical background
- Proficiency with algorithms
- Critical thinking and problem-solving skills

Course Objectives:

1. To introduce the ideas of soft computational techniques based on human experience.
2. To generate an ability to design, analyze and perform experiments on real life problems using various Neural Learning Algorithms.
3. To conceptualize fuzzy logic and its implementation for various real-world applications.

Learning Outcomes:

After completion of this course students will be able to -

1. apply soft computing techniques in various application.
2. to compare and choose appropriate NN architecture by analyzing various neural network architectures.
3. understand perceptron and counter propagation networks, can define the fuzzy systems.

Unit 1: Introduction to Soft Computing

02

- 1.1 Neural Networks: Definition, Advantages,
- 1.2 Applications, Scope.
- 1.3 Fuzzy logic: Definition, Applications. Genetic Algorithms: Definition, Applications

Unit 2: Neural Network

15

- 2.1 Fundamental Concept: Artificial Neural Network, Biological Neural Network
- 2.2 Brain vs. Computer-Comparison Between Biological Neuron and Artificial Neuron (Brain vs. Computer)
- 2.3 Artificial Neurons
- 2.4 Neural Networks and Architectures: Neuron Abstraction, Neuron Single Functions, Mathematical Preliminaries, Neural Networks Defined,
- 2.5 Architectures: Feed forward and Feedback
- 2.6 Salient Properties of Neural Networks
- 2.7 Geometry of Binary Threshold Neurons and Their Networks: Pattern Recognition and Data Classification, Convex Sets, Convex Hulls and Linear Separability, Space of Boolean Functions, Binary Neurons are Pattern Dichotomizers
- 2.8 Non-linearly Separable Problems

- 2.9 Capacity of a Simple Threshold Logic
- 2.10 Neuron, Revisiting the XOR Problem,
- 2.11 Multilayer Networks
- 2.12 How Many Hidden Nodes are Enough?
- 2.13 Learning and Memory: An Anecdotal Introduction
 - 2.13.1 Long Term Memory,
 - 2.13.2 The Behavioral Approach to Learning
 - 2.13.3 The Molecular Problem of Memory
 - 2.13.4 Learning Algorithms
 - 2.13.5 Error Correction and Gradient Descent Rules
 - 2.13.6 Learning Objective for TLNs
 - 2.13.7 Pattern Space and Weight Space
 - 2.13.8 Linear Separability
 - 2.13.9 Hebb Network
 - 2.13.10 Perceptron Network
 - 2.13.11 α - Least Mean Square Learning

Unit 3: Fuzzy Set Theory

09

- 3.1 Brief Review of Conventional Set Theory
- 3.2 Introduction to Fuzzy Sets
- 3.3 Properties of Fuzzy Sets
- 3.4 Operations on Fuzzy Sets
- 3.5 Crisp Relation, Fuzzy Relation
- 3.6 Tolerance and equivalence relation
- 3.7 Fuzzy Tolerance and equivalence relation
- 3.8 Fuzzy Max-Min and Max-Product Composition
- 3.9 Membership Functions
- 3.10 Fuzzification, Defuzzification to crisp sets
- 3.11 λ -Cuts for fuzzy Relations
- 3.12 Fuzzy (Rule-Based) system
- 3.13 Graphical technique of inference
- 3.14 Membership value assignment-Intuition, Inference

Unit 4: Genetic Algorithms

04

- 4.1 What are Genetic Algorithms?
- 4.2 Why Genetic Algorithms
- 4.3 Traditional Optimization and Search Techniques
- 4.4 Simple GA
- 4.5 Terminologies and Operators in GA
- 4.6 Encoding, Selection, Crossover
- 4.7 Mutation, Search Termination
- 4.8 Constraints in GA

Reference Books:

1. Fuzzy Logic With Engineering Applications, Timothy Ross, Wiley Publication.
2. Introduction to Soft Computing, Deepa & Shivanandan, Wiley Publication.
3. Genetic Algorithms in Search, Optimization and Machine Learning, David E. Goldberg, Pearson Education.

CS-463-MJP : Lab Course Based on Soft Computing

(Credits- 02)

C / C++ / Java/ Octave

Assignment 1

Programs to perform fuzzy set operations.

Assignment 2

Programs to apply De Morgans laws.

Assignment 3

Programs to implement activation functions.

Assignment 4

Programs to implement concept such as Hebb's Rule, FFN.

CS-481-OJT: On Job Training

Total duration: 120 Hours (Credits-4)

Learning Objectives:

1. To provide students with practical, hands-on-experience in applying theoretical knowledge to real-world tasks
2. To help students develop and enhance their skills, problem solving abilities and work culture of the industry
3. To foster effective teamwork and collaboration skills
4. To encourage students to build and expand their professional network by interaction with experienced experts and mentors in industry

Course Outcomes: On Completion of this course, student will be able to -

- CO 1: Enhance the knowledge related to various tools and technologies used in industry
- CO 2: Improve the ability to solve complex problems independently and creatively
- CO 3: Effectively utilize critical thinking and analytical skills in tackling real world challenges
- CO 4: Effectively communicate and collaborate skills through interaction with team members and mentors.
- CO 5: Get an experience in working on projects or related working within industry
- CO 6: Develop the ability to document process, design, implementation and testing
- CO 7: Familiar with specific industry domain relevant to internship
- CO 8: Complete projects and tasks as per the predetermined objectives

Guidelines for On Job Training (OJT)

1. Student must start the OJT/Internship immediately after semester-V examination during the vacation.
2. Student are expected to complete the IT related work/project within 120 hours assigned by organization (company/ industry/ consultancy/ institution).
3. The internship work may involve the IT related assignment(s) OR the maintenance of existing project OR the design/development of new project OR equivalent work.
4. College should assign the mentors/guides for students to monitor the progress throughout the OJT.
5. Students have to submit the weekly progress report duly signed by the concern authorities of organization to the assigned mentor.
6. At the end of OJT, students should prepare the documentation and submit a report to the college in prescribed format.
7. After completion, the final presentation and documentation will be evaluated by the examination panel.

Evaluation Pattern

The internal and external evaluation will be 50-50%

For all the courses, which are of four credits, total marks will be 100. Out of 50 marks will be allotted for internal evaluation and 50 marks for external evaluation.

For all the courses, which are of two credits, total marks will be 50. Out of 25 marks will be allotted for internal evaluation and 25 marks for external evaluation.

Theory Courses of four credits :

- Internal evaluation will be of 50 marks for which 3 continuous evaluation exams of 15, 15 and 20 marks will be conducted
- External evaluation will be of 50 marks

Theory Courses of two credits :

- Internal evaluation will be of 25 marks for which 2 continuous evaluation exams of 15 and 10 marks will be conducted
- External evaluation will be of 25 marks

Practical Courses of four credits :

- Internal evaluation will be of 50 marks out of which 30 marks will be for assignment submissions done throughout the semester and a test/viva will be conducted for 20 marks
- External evaluation will be of 50 marks

Practical Courses of two credits :

- Internal evaluation will be of 25 marks out of which 15 marks will be for assignment submissions done throughout the semester and a test/viva will be conducted for 10 marks
- External evaluation will be of 25 marks

Methods of assessment for internal evaluation:

Seminar, objective test, open book test, Quiz, viva, projects, assignments, group discussion, research paper review, case study, industrial visit

Passing percentage

The student must secure at least 40% marks of that course to earn the full credit.

Examination	Credits	Marks Out of	Passing marks (40%)
Internal	4	50	20
External	4	50	20
Internal	2	25	10
External	2	25	10

Note: There is separate passing for internal and external examination