Savitribai Phule Pune University

(Formerly University of Pune)

Two Year Degree Program in Computer Science

(Faculty of Science & Technology)

Revised Syllabi for

M.Sc. (Computer Science) Part-I

(For Colleges Affiliated to Savitribai Phule Pune University)

Choice Based Credit System Syllabus

To be implemented from Academic Year 2019-2020

Title of the Course: M.Sc. (Computer Science)

Preamble:

This syllabus is the extension of the existing syllabus which is currently being taught to M.Sc. (Computer Science) of Savitribai Phule Pune University for the last few years, but modified to be placed within the credit based system to be implemented from the academic year 2019-2020. However, there are few changes incorporated in the existing syllabus.

It is believed that the proposed changes as part of the credit based system will bring a qualitative change in the way M.Sc. (Computer Science) is taught, which will offer a more enriched learning experience. It aims to provide technology-oriented students with the knowledge and ability to develop creative solutions, and better understand the effects of future developments of computer systems and technology on people and society.

The syllabus is about developing skills to learn new technology, grasping the concepts and issues behind its use and the use of computers.

Course Structure:

Year/ Sem	Course Type	Course Code	Course Name	Credit	% of Assessment		
					IA	UE	Total
I Year Sem-I	Core Compulsory Theory Paper	CSUT111	Paradigm of Programming Language	4	30	70	100
		CSUT112	Design and Analysis of Algorithms	4	30	70	100
		CSUT113	Database Technologies	4	30	70	100
	Choice Based Optional Paper	CSDT114	Cloud computing	2	15	35	50
		CSDP114	Cloud Computing Practical	2	15	35	50
		OR					
		CSDT114	Artificial Intelligence	2	15	35	50
		CSDP114	Artificial Intelligence Practical	2	15	35	50
		OR					
		CSDT114	Web Services	2	15	35	50
		CSDP114	Web Services Practical	2	15	35	50
	Core Compulsory Practical Paper	CSUP115	PPL and Database Technologies Practical	4	30	70	100

Year/ Sem	Course Type	Course Code	Course Name	Credit	% of Assessment		
					IA	UE	Total
I Year Sem-II	Core Compulsory Theory Paper	CSUT121	Advanced Operating System	4	30	70	100
		CSUT122	Mobile Technologies	4	30	70	100
		CSUT123	Software Project Management	4	30	70	100
	Choice Based Optional Paper	CSDT124	Project	2	15	35	50
		CSDP124	Project related Assignments	2	15	35	50
		OR					
		CSDT124	Human Computer Interaction	2	15	35	50
		CSDP124	Human Computer Interaction Practical	2	15	35	50
		OR					
		CSDT124	Soft Computing	2	15	35	50
		CSDP124	Soft Computing Practical	2	15	35	50
	Core Compulsory Practical Paper	CSUP125	Practical on Advanced OS & Mobile Technologies	4	30	70	100

Year/ Sem	Course Type	Course Code	Course Name	Credit	% of Assessment		
					IA	UE	Total
II Year Sem-III	Core Compulsory Theory Paper	CSUT231	Software Architecture and Design Pattern	4	30	70	100
		CSUT232	Machine Learning	4	30	70	100
		CSUT233	Evolutionary Algorithms	4	30	70	100
	Choice Based Optional Paper	CSDT234	Big Data	2	15	35	50
		CSDP234	Big Data Practical	2	15	35	50
		OR					
		CSDT234	Web Analytics	2	15	35	50
		CSDP234	Web Analytics Practical	2	15	35	50
		OR					
		CSDT234	Project	2	15	35	50
		CSDP234	Project related Assignments	2	15	35	50
	Core Compulsory Practical Paper	CSUP235	Practical on Software Architecture and Design Pattern and Machine Learning	4	30	70	100

Year/ Sem	Subject	Paper	Title of Paper	Credit	% of Assessment		
					IA	UE	Total
II Year Sem-IV	Core	CSUIT241	Industrial Training /Institutional project	20			

IA :- Internal Assessment, UE :- University Examination

Equivalence of Previous Syllabus:

Old Subject	New Subject
Principles of Programming Languages	Paradigm of Programming Language
Advanced Networking	No Equivalence
Distributed Database Concepts	Database Technologies
Design and Analysis of Algorithms	Design and Analysis of Algorithms
Network Programming	No Equivalence
Digital Image Processing	No Equivalence
Advanced Operating Systems	Advanced Operating Systems
Data Mining and Data Warehousing	Big Data
Project	Project
Programming With DOT NET	No Equivalence
Artificial Intelligence	Artificial Intelligence
Advance Design and Analysis of Algorithms	Evolutionary Algorithms
Software Metrics & Project Management	Software Project Management
Mobile Computing	Mobile Technologies
Soft Computing	Soft Computing
Project	Project
Web Services	Web Services
Database and System Administrator	No Equivalence
Functional Programming	No Equivalence
Business Intelligence	No Equivalence
Industrial Training /Institutional project	Industrial Training /Institutional project
Parallel Computing	No Equivalence
Embedded System	No Equivalence
Software Quality Assurance	No Equivalence
Modeling and Simulation	No Equivalence

Detailed Syllabus:

Course Code: CSUT111	**Course Name: Paradigm of Programming Language**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 4**
Course Prerequisites:	Student should have basic knowledge of: • Procedural Language like C • Object-Oriented Languages (C++ and Java) • Concepts of Operating Systems • Basic Data Structures and Algorithms.	
Course Objectives:	To Prepare student to think about programming languages analytically: • Separate syntax from semantics • Compare programming language designs • Understand their strengths and weaknesses • Learn new languages more quickly • Understand basic language implementation techniques • Learn small programs in different programming Languages	
Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction** • The Art of Language Design • The Programming Language Spectrum • Why Study Programming Languages? • Compilation and Interpretation • Programming Environments	2
2	**Names, Scopes, and Bindings** • The Notion of Binding Time • Object Lifetime and Storage Management • Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection Scope Rules • Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping The meaning of Names in a Scope • Aliases, Overloading, Polymorphism and Related Concepts, the Binding of Referencing Environments • Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures Macro Expansion •	5

3	**Control Flow** • Expression Evaluation , Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation • Structured and Unstructured Flow, Structured Alternatives to goto • Sequencing • Selection - Short-Circuited Conditions, Case/Switch Statements Iteration • Iteration - Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion • Recursion - Iteration and Recursion, Applicative- and Normal-Order Evaluation	5
4	**Data Types** • Introduction • Primitive Data Types • Numeric Types : Integer, Floating point, Complex , Decimal, Boolean Types, Character Types • Character String Types • Design Issues, Strings and Their Operations, String Length Operations, Evaluation, Implementation of Character String Types • User defined Ordinal types Enumeration types, Designs Evaluation Subrange types, Ada's design Evaluation Implementation of user defined ordinal types • Array types • 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 • Associative Arrays • Structure and operations, Implementing associative arrays, • Record types • Definitions of records, References to record fields, Operations on records, Evaluation, Implementation of Record types • Union Types • Design issues, Discriminated versus Free unions, Evaluation, Implementation of Union types	8

	• Pointer and Reference Types • Design issues, Pointer operations, Pointer problems, Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation • Implementation of pointer and reference types - Representation of pointers and references Solution to dangling pointer problem Heap management	
5	**Subprograms and Implementing Subprograms** • Introduction • Fundamentals of Subprograms • Design Issues for subprograms • Local Referencing Environments • Parameter-Passing Methods • Parameters That Are • Subprograms • Overloaded Subprograms • Generic Subroutines, Generic Functions in C++, Generic Methods in Java • Design Issues for Functions • User-Defined Overloaded Operators • Coroutines • Implementing Subprograms • The General Semantics of Calls and Returns • Implementing "Simple" Subprograms • Implementing Subprograms with Stack-Dynamic Local Variables • Nested Subprograms • Blocks • Implementing Dynamic Scoping	5
6	**Data Abstraction and Object Orientation** • Object-Oriented Programming • Encapsulation and Inheritance Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance • Initialization and Finalization Choosing a Constructor, References and Values, Execution Order, Garbage Collection • Dynamic Method Binding • Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures • Multiple Inheritance • Semantic Ambiguities, Replicated Inheritance,	8

	Shared Inheritance, Mix-In Inheritance	
7	**Concurrency** • Introduction : Multiprocessor Architecture Categories of concurrency, Motivations for studying concurrency • Introduction to Subprogram-level, concurrency Fundamental concepts, Language Design for concurrency, Design Issues • Semaphores - Introduction Cooperation synchronization, Competition Synchronization, Evaluation • Monitors - Introduction, Cooperation synchronization, Competition Synchronization, Evaluation, • Message Passing Introduction- The concept of Synchronous Message Passing • Java Threads - The Thread class –Priorities, Competition Synchronization Cooperation Synchronization, Evaluation	5
8	**Functional Programming in Scala** • Strings • Numbers • Control Structures • Classes and Properties • Methods • Objects • Functional Programming • List, Array, Map, Set	10

References:

Sr. No.	Title of the Book	Author/s	Publication
1	Programming Language Pragmatics, 3e	Michel L. Scott	Kaufmann Publishers, An Imprint of Elsevier, USA
2	Concepts of Programming Languages, Eighth Edition	Robert W. Sebesta	Pearson Education
3	Scala Cookbook	Alvin Alexander	O'REILLY publication

Course Code: CSUT112	**Course Name: Design and Analysis of Algorithm**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 4**
Course Prerequisites:	• Basic knowledge of algorithms and programming concepts • Data Structures and Advanced Data Structures • Basic Knowledge of Graphs and Algorithms	
Course Objectives:	• To design the algorithms • To select the appropriate algorithm by doing necessary analysis of algorithms • To learn basic Algorithm Analysis techniques and understand the use of asymptotic notation • Understand different design strategies • Understand the use of data structures in improving algorithm performance • Understand classical problem and solutions • Learn a variety of useful algorithms • Understand classification of problems • To provide foundation in algorithm design and analysis • To develop ability to understand and design algorithms in context of space and time complexity.	
Chapter	**Course Contents**	**No. of Lectures**
1	**Basics of Algorithms** • Algorithm definition and characteristics • Space complexity • Time complexity, worst case-best case-average case • complexity, asymptotic notation • Recursive and non-recursive algorithms • Sorting algorithms (insertion sort, heap sort, bubble sort) • Sorting in linear time: counting sort, concept of bucket and radix sort • Searching algorithms: Linear, Binary	8
2	**Divide and conquer strategy** • General method, control abstraction • Binary search • Merge sort, Quick sort • Comparison between Traditional Method of Matrix Multiplication vs. Strassen's Matrix Multiplication	5

3	**Dynamic Programming** • Principle of optimality • Matrix chain multiplication • 0/1 Knapsack Problem i)Merge & Purge ii)Functional Method • Concept of Shortest Path • Single Source shortest path i)Dijkstra's Algorithm ii)Bellman Ford Algorithm • All pairs Shortest Path • Floyd- Warshall Algorithm • Longest common subsequence, • String editing, Travelling Salesperson problem	10
4	**Greedy Method** • Job sequencing with deadlines, • Minimum-cost spanning trees: Kruskal and Prim's algorithm • Optimal storage on tapes • Optimal merge patterns • Huffman coding • Shortest Path: Dijkstra's algorithm	7
5	**Decrease and Conquer** • Definition of Graph Representation of Graph • By Constant - DFS and BFS • Topological sorting • Connected components and spanning trees • By Variable Size decrease Euclid's algorithm • Flow in graph • Articulation Point and Bridge edge	5
6	**Backtracking** • General method • Fixed Tuple vs. Variable Tuple Formulation • n- Queen's problem • Graph coloring problem • Hamiltonian cycle • Sum of subsets	5
7	**Branch and Bound** • Introduction • Definitions of LCBB Search • Bounding Function, Ranking Function • FIFO BB Search • Traveling Salesman problem Using Variable tuple	5

	• Formulation using LCBB • 0/1 knapsack problem using LCBB	
8	**Problem Classification** • Nondeterministic algorithm • The class of P, NP, NP-hard and NP - Complete problems • Cook's theorem	3

References:

Sr. No.	Title of the Book	Author/s	Publication
1	Computer algorithms	Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran	Galgotia Publication
2	T. Cormen, C. Leiserson, & R. Rivest	Algorithms	MIT Press
3	A. Aho, J. Hopcroft & J. Ullman	The Design and Analysis of Computer Algorithms	Addison Wesley
4	Donald Knuth	The Art of Computer Programming	Addison Wesley
5	Steven Skiena	The Algorithm Manual	Springer
6	Jungnickel	Graphs, Networks and Algorithms	Springer

Course Code: CSUT113	**Course Name: Database Technologies**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 4**
Course Prerequisites:	• Knowledge of file system concepts • Strong foundation of Related database Concepts (Basic & Advanced) • A firm foundation of any RDBMS package	
Course Objectives:	• Provide an overview of the concept of NoSQL technology. • Provide an insight to the different types of NoSQL databases • Make the student capable of making a choice of what database technologies to use, based on their application needs.	

Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction to NOSQL (Core concepts)**	18
	Why NoSQL	
	Aggregate Data Models	
	Data modeling details	
	Distribution Models	
	Consistency	
	Version stamps	
	Map-Reduce	
2	**Implementation with NOSQL databases**	15
	Key-Value Databases (Riak)	
	Document Databases (Mongodb)	
	Column-Family stores (Cassandra)	
	Graph databases (Neo4j)	
3	**Schema Migrations**	5
4	**Polygot Persistence (Multi model types)**	5
5	**Beyond NoSQL**	3
6	**Choosing your database**	3

References:

Sr. No.	**Title of the Book**	**Author/s**	**Publication**
1	NoSQL Distilled	Pramod Sadalge, Martin Fowler	
2	NoSQL for Dummies	A Willy Brand	
3	http://nosql-database.org		

Course Code: CSDT114	Course Name: Cloud Computing	Total Lectures (48 Hours)
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 2**
Course Prerequisites:	• Operating System • Fundamentals of Computer Networks • Good Understanding of Object Oriented Programming Concepts	
Course Objectives:	• To understand the principles and paradigm of Cloud Computing • To appreciate the role of Virtualization Technologies • Ability to design and deploy Cloud Infrastructure • Understand cloud security issues and solutions	

Chapter	Course Contents	No. of Lectures
1	**Introduction to Cloud Computing** Overview, Roots of Cloud Computing, Layers and Types of Cloud, Desired Features of a Cloud, Benefits and Disadvantages of Cloud Computing, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, Challenges and Risks. Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Center Technology, Virtualization Technology, Web Technology, Multitenant Technology, Service Technology.	10
2	**Architecture, Services and Applications** Exploring the Cloud Computing Stack, Connecting to the Cloud, Infrastructure as a Service, Platform as a Service, Saas Vs. Paas, Using PaaS Application Frameworks, Software as a Service Cloud Deployment Models, Public vs Private Cloud, Cloud Solutions, Cloud ecosystem, Service management, Computing on demand, Identity as a Service, Compliance as a Service Future of cloud computing and Emerging trends.	10
3	**Abstraction and Virtualization** Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Understanding Machine Imaging, Porting Applications, Virtual Machines	

	Provisioning and Manageability Virtual Machine Migration Services, Virtual Machine Provisioning and Migration in Action, Provisioning in the Cloud Context Virtualization of CPU, Memory , I/O Devices, Virtual Clusters and Resource management, Virtualization for Data Center Automation	8
4	**Programming, Environments and Applications** Features of Cloud and Grid Platforms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments, Understanding Core OpenStack Ecosystem. Applications: Moving application to cloud, Microsoft Cloud Services, Google Cloud Applications, Amazon Cloud Services, Cloud Applications (Social Networking, E-mail, Office Services, Google Apps, Customer Relationship Management).	10
5	**Security In The Cloud** Security Overview – Cloud Security Challenges and Risks – Software-as-a-Service Security – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control – Autonomic Security. Autonomic Security Storage Area Networks, Disaster Recovery in Clouds.	10

References:

Sr. No.	Title of the Book	Author/s	Publication
1	Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center	Brian J.S. Chee and Curtis Franklin	CRC Press, ISBN :9781439806128
2	Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi	Mastering Cloud Computing: Foundations and Applications Programming	McGraw Hill, ISBN: 978 1259029950, 1259029956
3	Kai Hwang, Geoffrey C Fox, Jack G Dongarra	Distributed and Cloud Computing, From Parallel Processing to the Internet of Things	Morgan Kaufmann Publishers, 2012.

CSDP114: Cloud Computing Practical Assignments

Sr. No	Assignment
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 services.
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.

Course Code:	**Course Name: Artificial Intelligence**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme:** **IA: 30 Marks** **UE: 70 Marks**	**No. of Credits**
Course Prerequisites:	• Concepts of Data structures and Design and Analysis of algorithms. • Strong data analytics skills. • Strong will to learn machine learning languages.	
Course Objectives:	• To learn various types of algorithms useful in Artificial Intelligence (AI). • To convey the ideas in AI research and programming language related to emerging technology. • To understand the numerous applications and huge possibilities in the field of AI that goes beyond the normal human imagination.	
Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction to Artificial Intelligence:** Introduction and Intelligent systems, What Is AI, The Foundations of Artificial Intelligence, The History of Artificial Intelligence, Applications of AI, Early work in AI and related fields, AI problems and Techniques.	**2**
2	**Searching:** -Defining AI problems as a State Space Search: example, Search and Control Strategies, Problem Characteristics, Issues in Design of Search Programs, Production System. Blind Search Techniques : -BFS, DFS, DLS, Iterative Deepening, Search, Bidirectional Search, Uniform cost Search. Heuristic search techniques: -Generate and test ,Hill Climbing, Best First search, Constraint Satisfaction, Mean-End Analysis, A*,AO*.	**10**

3	**Knowledge Representation:** Representations and Mappings, Approaches to Knowledge Representation, Knowledge representation method, Propositional Logic, Predicate logic, Representing Simple facts in Logic, Resolution, Forward and backward chaining . Knowledge Representation Structure-Weak Structures, Strong Structures. Semantic Networks, Frames, Conceptual Dependencies, Scripts. Game Playing- Minimax Search Procedures, Adding alpha-beta cutoffs.	**15**
4	**Introduction to AI with Python:** Introduction to Python , why python with AI, Features of Python, Basics of Python, Python statements, Methods & Functions using python, Basic and advanced modules &Packages, Python Decorators and generators .Advanced Objects & Data structures.	**13**
5	**Machine Learning:** Why Machine learning, Types of Machine Learning: Supervised learning- Classification & Regression. Decision tree, Random Forest, KNN, Logistic algorithms. Unsupervised learning-Clustering & Association. K-means for clustering, Apriori algorithm. Support Vector Machine (SVM),Reinforcement learning.	**8**

References:

Sr. No.	Title of the Book	Author/s	Publication
1	Computational Intelligence	Eberhart	Elsevier Publication
2	Artificial Intelligence: A New Synthesis	Nilsson	Elsevier Publication
3	Artificial Intelligence with Python	PrateekJoshi	Packt Publishing Ltd
4	Reinforcement and Systematic Machine Learning for Decision Making,	Parag Kulkarni	Wiley-IEEE Press Edition
5	Artificial Intelligence	Saroj Kausik	Cengage Learning
6	Introduction to Machine Learning	EthemAlpaydin	PHI 2nd Edition

CSDP114: Artificial Intelligence Practical

Sr. No.	Assignment
1	Subject teacher should conduct first lab practical on basic programs using python for introducing and using python environment such as, a) Program to print multiplication table for given no. b) Program to check whether the given no is prime or not. c) Program to find factorial of the given no and similar programs.
2	Write a program to implement List Operations(Nested list, Length, Concatenation, Membership ,Iteration ,Indexing and Slicing), List Methods(Add, Append, Extend & Delete)
3	Write a program to Illustrate Different Set Operations.
4	Write a program to implement Simple Chatbot.
5	Write a program to implement Breadth First Search Traversal.
6	Write a program to implement Depth First Search Traversal.
7	Write a program to implement Water Jug Problem.
8	Write aprogram to implement K -Nearest Neighbor algorithm.
9	Write a program to implement Regression algorithm.
10	Write a program to implement Random Forest Algorithm.

Course Code: CSDT 114	**Course Name: Web Services**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 2**
Course Prerequisites:	• Strong knowledge about Java programming. • Good Understanding of Object Oriented Programming concepts. • Must be familiar with XML.	
Course Objectives:	• To understand the details of web services technologies like WSDL,UDDI, SOAP • To learn how to implement and deploy web service client and server • To explore interoperability between different frameworks • To understand the concept of RESTful system.	
Chapter	**Course Contents**	**No. of Lectures**
1	**Web Service and SOA fundamentals** **Evolution and Emergence of Web Services** – Evolution of distributed computing, Core distributed computing technologies — client/server, CORBA, JAVA RMI, Microsoft DCOM, Challenges in Distributed Computing, role of J2EE and XML in distributed computing, emergence of Web Services and Service Oriented Architecture (SOA). **Introduction to Web Services** — The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services. **Web Services Architecture** — Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication models, basic steps of implementing web services.	12
2	**SOAP: Simple Object Access Protocol** Inter-application communication and wire protocols, SOAP as a messaging protocol, Structure of a SOAP message, SOAP communication model, Building SOAP Web Services, developing SOAP Web Services using Java, Error handling in SOAP, Advantages and disadvantages of SOAP.	12

3	**Unit III : Describing and Discovering Web Services** **WSDL -** WSDL in the world of Web Services, Web Services life cycle, anatomy of WSDL definition document, WSDL bindings, WSDL Tools, limitations of WSDL, Service discovery, role of service discovery in a SOA, service discovery mechanisms, **UDDI** – UDDI Registries, uses of UDDI Registry, Programming with UDDI, UDDI data structures, support for categorization in UDDI Registries, Publishing API, Publishing information to a UDDI Registry, searching information in a UDDI Registry, deleting information in a UDDI Registry, limitations of UDDI.	12
4	**Unit IV : The REST Architectural style :** Introducing HTTP, The core architectural elements of a RESTful system, Description and discovery of RESTful web services, Java tools and frameworks for building RESTful web services, JSON message format and tools and frameworks around JSON, Build RESTful web services with JAX-RS APIs, The Description and Discovery of RESTful Web Services, Design guidelines for building RESTful web services, Secure RESTful web services	12

References:

Sr. No.	Title of the Book	Author/s	Publication
1	Building Web Services with Java, 2nd Edition	S. Graham and others	Pearson Edn., 2008.
2	J2EE Web Services	Richard Monson-Haefel	Pearson Education.
3	Java Web Services Programming,	R.Mogha,V.V.Preetham	Wiley India Pvt.Ltd.
4	XML, Web Services, and the Data Revolution	F.P.Coyle	Pearson Education

CSDP114: Web Services Practical Assignments

Pre-requisites

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

Objectives

- To understand how to develop web services using Java/PHP/.Net

Sr. No.	Assignment
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 user name 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 given number and create 'Dynamic Web Project', which will host the client application that will send 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 user name and password (use database for storing username and password) and create 'Dynamic Web Project', which will host the client application that will send user name and password and test the web service.
6.	Create 'Dynamic Web Project', which will host your web service functionality to select employee details (use 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 amount in Rupees & type of conversion and tests the web service.

10.	Create 'Dynamic Web Project', which will host your web service functionality to give the suggestion for given key word and create 'Dynamic Web Project', which will host the client application that tests the web service.
11.	Create 'Dynamic Web Project', which will host your web service functionality to find area and volume of the circle and create 'Dynamic Web Project', which will host the client application that tests the web service.
12.	Create 'Dynamic Web Project', which will host your web service functionality to find number of vowels in the given string and create 'Dynamic Web Project', which will host the client application that tests the web service.
13.	Create 'Dynamic Web Project', which will host your web service functionality to convert decimal number to Binary, Octal, Hexa Decimal and create 'Dynamic Web Project', which will host the client application that will send decimal number & type of conversion and test the web service.
14.	Create 'Dynamic Web Project', which will host your web service functionality to validate user name and password (use database for storing username and password) and create 'Dynamic Web Project', which will host the client application that will send user name and password and test the web service.
15.	Create 'Dynamic Web Project', which will host your web service functionality for returning book price and create 'Dynamic Web Project', which will host the client application that will send Book Name

CSUP115: PPL and Database Technologies Practical

LIST OF SCALA PROGRAMS (PPL)

Control Structures

1. Write a program to calculate average of all numbers between n1 and n2(eg.100 to 300 Read values of n1 and n2 from user)
2. Write a program to calculate factorial of a number.
3. Write a program to read five random numbers and check that random numbers are perfect number or not.
4. Write a program to find second maximum number of four given numbers.
5. Write a program to calculate sum of prime numbers between 1 to 100
6. Write a program to read an integer from user and convert it to binary and octal using user defined functions.

Arrays

1. Write a program to find maximum and minimum of an array
2. Write a program to calculate transpose of a matrix.
3. Write a program to calculate determinant of a matrix,
4. Write a program to check if the matrix is upper triangular or not.
5. Write a program to sort the matrix using insertion sort.
6. Write a program for multiplication of two matrices(Validate number of rows and columns before multiplication and give appropriate message)

String

1. Write a program to count uppercase letters in a string and convert it to lowercase and display the new string.
2. Write a program to read a character from user and count the number of occurrences of that character.
3. Write a program to read two strings. Remove the occurrence of second string in first string.
4. Create array of strings and read a string from user. Display all the elements of array containing given string.

Classes and Objects

1. Define a class CurrentAccount (accNo, name, balance, minBalance). Define appropriate constructors and operations withdraw(), deposit(), viewBalance(). Create an object and perform operations.
2. Define a class Employee (id, name, salary). Define methods accept() and display(). Display details of employee having maximum salary.
3. Create abstract class Order (id, description). Derive two classes PurchaseOrder& SalesOrder with members Vendor and Customer. Create object of each PurchaseOrder and SalesOrder. Display the details of each account.
4. Create abstract class Shape with abstract functions volume() and display(). Extend two classes Cube and Cylinder from it. Calculate volume of each and display it.

5. Create class Project (id, name, location). Define parameterized constructor. Keep a count of each object created and display the details of each project.
6. Define a class Sports (id, name, description, amount). Derive two classes Indoor and Outdoor. Define appropriate constructors and operations. Create an object and perform operations.
7. Design abstract class Employee with computeSal() as abstract function. Create two subclasses Worker and Manager. Salary of worker should be calculated on hourly basis of work and Salary of Manager should be calculated on monthly basis with additional incentives.

List

1. Create Lists using five different methods(Lisp style , Java style, fill, range and tabulate methods)
2. Create two Lists and Merge it and store the sorted in ascending order.
3. Create a list of integers divisible by 3 from List containing numbers from 1 to 50.
4. Create a list of even numbers up to 10 and calculate its product.
5. Write a program to create list with 10 members using function $3n^2+4n+6$
6. Write a program to create a list of 1 to 100 numbers. Create second list from first list selecting numbers multiple of 10.
7. Create a list of 50 members using function 2n+3. Create second list excluding all elements multiple of 7.

Map

1. Write a user defined functions to convert lowercase letter to uppercase and call the function using Map.
2. Write a program to create map with Rollno and FirstName. Print all student information with same FirstName.

Set

1. Write a program to create two sets and find common elements between them.
2. Write a program to display largest and smallest element of the Set
3. Write a program to merge two sets and calculate product and average of all elements of the Set

Course Code: CSUT121	**Course Name: Advanced Operating System**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 4**
Course Prerequisites:	• Working knowledge of C programming. • Basic Computer Architecture concepts. • Basic algorithms and data structure concepts.	
Course Objectives:	This course teaches Advanced Operating Systems Concepts using Unix/Linux. This course strikes a delicate balance between theory and practical applications In fact, most Units start with the theory and then switches focus on how the concepts are implemented in a C program. This course describes the programming interface to the Unix/Linux system - the system call interface. It is intended for anyone writing C programs that run under Unix/Linux. This course provides an understanding of the functions of Operating Systems. It also provides provide an insight into functional modules of Operating Systems. It discusses the concepts underlying in the design and implementation of Operating Systems.	
Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction to UNIX/Linux Kernel** • System Structure, User Perspective, Assumptions about Hardware, Architecture of UNIX Operating System (TextBook-1: Chapter Topics: 1.2, 1.3, 1.5, 2.1) • Concepts of Linux Programming- Files and the Filesystem, Processes, Users and Groups, Permissions, Signals, Interprocess Communication (TextBook-3: Chapter 1- relevant topics)	04
2	**File and Directory I/O** • Buffer headers, structure of the buffer pool, scenarios for retrieval of a buffer, reading and writing disk blocks, inodes, structure of regular file, open, read, write, lseek, close, pipes, dup (TextBook- 1: Chapter Topics: 3.1-3.4, 4.1, 4.2, 5.1-5.3, 5.5-5.7, 5.12, 5.13) • open, creat, file sharing, atomic operations, dup2, sync, fsync, and fdatasync, fcntl, /dev/fd, stat, fstat, lstat, file types, Set-User-ID and Set-Group-ID, file access permissions, ownership of new files and directories, access function, umask function, chmod and fchmod, sticky bit, chown, fchown, and lchown, file size, file truncation, file systems, link, unlink, remove, and rename functions, symbolic links, symlink and readlink functions, file times, utime, mkdir and rmdir, reading directories, chdir, fchdir, and getcwd, device special files (TextBook-2: Chapter Topics: 3.3, 3.4, 3.10-3.14, 3.16, 4.2-4.23)	15

3	**Process Environment, Process Control and Process Relationships** • Process states and transitions, layout of system memory, the context of a process, saving the context of a process, sleep, 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, Process Scheduling (TextBook-1: Chapter Topics: 6.1-6.4, 6.6, 7.1-7.8, 8.1) • Process termination, environment list, memory layout of a C program, shared libraries, environment variables, setjmp and longjmp, getrlimit and setrlimit, process identifiers, fork, vfork, exit, wait and waitpid, waitid, wait3 and wait4, race conditions, exec, changing user IDs and group IDs,system function, user identification, process times (TextBook-2: Chapter Topics: 7.3, 7.5-7.7, 7.9-7.11, 8.2-8.11, 8.13, 8.15, 8.16)	15
4	**Memory Management** • 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, Opportunistic Allocation (TextBook-3: Chapter 8) • Swapping, Demand Paging (TextBook-1: Chapter Topics: 9.1, 9.2)	06
5	**Signal Handling** • Signal concepts, signal function, unreliable signals, interrupted system calls, reentrant functions, SIGCLD semantics, reliable-signal technology, kill and raise, alarm and pause, signal sets, sigprocmask, sigpending, sigsetjmp and siglongjmp, sigsuspend, abort, system function revisited, sleep (TextBook-2: Topics: 10.2-10.13, 10.15-10.19)	08

References:

Sr. No.	Title of the Book	Author/s	Publication
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

Course Code: CSUT122	**Course Name: Mobile Technologies**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme:** **IA: 30 Marks** **UE: 70 Marks**	**No. of Credits 4**
Course Prerequisites:	• Concepts of Networking • Conversant with OS internals • Familiar with the network Protocol stack • Gain knowledge about different mobile platform and application development • Brief History of wireless communication	
Course Objectives:	• To impart basic understanding of the wireless communication systems. • To expose students to various aspects of mobile and ad-hoc networks. • Understand the issues relating to Wireless applications • Understand the Mobile security	
Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction to Mobile Computing** • Introduction and need for Mobile computing • Mobility and portability • Mobile and Wireless devices • Mobile Applications • Mobile Operating system – IOS, BlackBery, Windows phone, Plam OS, Symbian OS,PhoneGap	03
2	**Android Fundamentals** • Introduction to Android - Overview and evolution of Android , Features of Android, Android architecture • Components of an Android Application, Manifest file • Android Activity • Service Lifecycle	07
3	**Android UI Design** • Basic UI Designing (Form widgets ,Text Fields , Layouts ,[dip, dp, sip, sp] versus px) • Intent(in detail) • All components (e.g Button , Slider, Image view, Toast) Event Handling • Adapters and Widgets • Menu	07

4	**Android Thread and Notification** • Threads running on UI thread (runOnUiThread) • Worker thread • Handlers & Runnable • AsynTask (in detail) • Broadcast Receivers • Services and notifications • Toast • Alarms	07
5	**Advanced Android Programming** • Content Providers – SQLite Programming • JSON Parsing • Accessing Phone Service(Call, SMS, MMS) • Location based services	05
6	**PhoneGap Programming** • Why Use PhoneGap? • How PhoneGap Works • Designing for the Container • Writing PhoneGap Applications • Building PhoneGap Applications • PhoneGap Limitations • PhoneGap Plug-Ins • Hello, World! Program • PhoneGap APIs –1 **Accelerometer:** • Querying Device Orientation, • Watching a Device's Orientation, • Creating a Contact, Searching for Contacts, Cloning Contacts, Removing Contacts.	12
7	**iOS Fundamentals** • **Introduction** - What is IOS ,IOS Architecture, Frameworks, Application Life Cycle, Features • **Swift** - Introduction to Swift ,General Concepts of Swift • **Xcode** - Introduction to Xcode , Navigator, Editor Utility, Tools, Console, Document, Simulator, Instruments • **Startup** - Application Templates, Introduction to Storyboard , Hello World Application, How 'Hello World' Working, Debugging Database, Plist, Preference, Sqlite Web Service, Restful Web Service (JSON & XML)	08

References:

Sr. No.	Title of the Book	Author/s	Publication
1	A Course in Machine Learning	Hal Daumé III	
2	IOS Apprentice	Matthijs Hollemans	
3	PhoneGap: Beginner's Guide	Giorgio Natili, Purusothaman Ramanujam	PACKT Publication
4	Beginning Android Application Development	Wei-Meng Lee Wiley	

Course Code: CSUT123	**Course Name: Software Project Management**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 4**
Course Prerequisites:	• Software Engineering • Basic testing concepts	
Course Objectives:	• Software Metrics and Project Management covers skills that are required to ensure successful medium and large scale software projects. • It examines Requirements Elicitation, Project Management, Verification &Validation and Management of Large Software Engineering Projects. • Students learn to select and apply project management techniques for process modeling, planning, estimation, process metrics and risk management; perform software verification and validation using inspections, design and execution of system test cases.	
Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction to Project Management** • What is a Project? • What is Project management? • Project phases and project life cycle • Organizational structure • Qualities of Project Manager • WBS	4
2	**Project Management Components** • Project Integration Management-Project plan development and execution • Change controls • CCB • Configuration management	6
3	**Scope Management** • Strategic planning • Scope planning, definition • Verification and control	4
4	**Time management** • Activity planning • Schedule development and control • GANTT Chart	2
5	**Cost Management** • Cost estimation and Control • COCOMO model • BASIC COCOMO NUMERICALS	2
6	**Quality Management** • Quality planning and assurance	2

7	**Human Resource Management** • Organizational planning • Staff acquisition	2
8	**Communication Management** • Information distribution • Reporting	2
9	**Risk Management** • Risk identification • Quantification and control	2
10	**Procurement Management** • Solicitation management and control • Contract administration	2
11	**Software Metrics** • The scope of software metrics • Size- oriented metrics • Function oriented • Software metrics data collection • Analyzing software data	6
12	**Software Reliability** • Measurement and prediction • Resource measurement • Productivity, teams and tools	6
13	**Planning a measurement program** • What is metrics plan? • Developing goals, questions and metrics • Where and When: Mapping measures to activities • How: Measurement tools • Who: Measurers , analyst, tools revision plans	4
14	**Quality Standards** • CMM levels • KPA's • PSP/TSP	4

References:

Sr. No.	Title of the Book	Author/s	Publication
1.	Software Engineering	Roger Pressman	McGraw-Hill
2.	Software Metrics for Project Management and process improvement	Robert B. Grady	Prentice hill

CSDT124: Project Guidelines

CSDP124: Project Related Assignments

Assignment 1

Assignment 2

Assignment 3

Assignment 4

Course Code: CSDT124	**Course Name: Human Computer Interaction**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme: IA: 30 Marks UE: 70 Marks**	**No. of Credits 2**
Course Prerequisites:	• 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:	• Design effective dialog for HCI. • Design effective HCI for individuals and persons with disabilities. • Assess the importance of user feedback. • Explain the HCI implications for designing multimedia/ ecommerce/ e-learning Web sites. • Develop meaningful user interface.	
Chapter	**Course Contents**	**No. of Lectures**
1	**FOUNDATIONS OF HCI** The Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms.	9
2	**DESIGN & SOFTWARE PROCESS** Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design	10
3	**MODELS AND THEORIES** Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration models-Hypertext, Multimedia and WWW.	10
4	**MOBILE HCI** Mobile Ecosystem: Platforms, Application frameworks Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.	10

5	**WEB INTERFACE DESIGN** Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow, Case Studies.	9

References:

Sr. No.	Title of the Book	Author/s	Publication
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

CSDP124: Human Computer Interaction Practical Assignments

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

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 behavior 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 behavior, problems, interests…

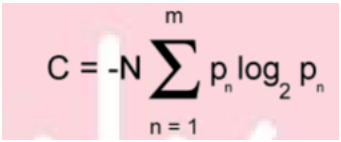
   Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

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
   1. Calculate Screen Complexity for existing Graphical User Interface (GUI).
   2. Redesign the Screen by applying various guidelines to lower the complexity of selected Graphical User Interface (GUI) to achieve simplicity

   Method for Measuring Complexity:
   1. Draw a rectangle around each element on a screen, including captions, controls, headings, data, title, and so on.
   2. Count the number of elements and horizontal alignment points (the number of columns in which a field, inscribed by a rectangle, starts).
   3. Count the number of elements and vertical alignment points (the number of rows in which an element, inscribed by a rectangle, starts).
   4. Calculate number of bits required by horizontal (column) alignment points and number of bits required by vertical (row) alignment points by applying following formula for calculating the measure of complexity.



   C, complexity of the system in bits
   N, total number of events (widths or heights)
   m, number of event classes (number of unique widths or heights)
   pn, probability of occurrence of the nth event class (based on the frequency of events within that class)
   5. Calculate overall complexity by adding the number bits required by horizontal alignment points and vertical alignment points.

5) Design/Redesign web user interface based on Gestalt theories and comment on the principle applied and justify. Alsoanalyze one image in which Gestalt principle is applied and comment.

   Example: Take a look at old IBM logo:

   You recognize the letters as an I, a B, and an M, no problem there. But they aren't letters at all; the whole thing is a compilation of bright blue horizontal lines arranged to create the perception of a set of letters. Gestalt Property used here is Closure. Closure means that we "close" objects that are themselves not complete; not only completing the figure in our

perception, but perceiving the figure as having an extra element of aesthetic design; we look for a simple, recognizable pattern.

6) Design an application which consists of different types of menus such as Menu bar, Pull-Down Menu, Cascading Menu, Pop-up Menus,Tear-off Menus. Apply and explain general menu design guidelines applied for formatting, ordering, phrasing, selecting choices, and navigating menus for application which is designed.

7) Implement different Kinds of Windows such as message boxes, palette Windows, Pop-up Windows, primary window, secondary window, dialog boxes, message box etc. For every window designed for the application explain:
- Purpose
- Description
- Components
- Kind window

8) Identify separate lines of business, e.g., medical, greeting cards, law etc. Design an application using proper guidelines for icons. Comment on design of icons and their relevance in the system.

Icon design is an important process. Meaningful and recognizable icons will speed learning and recall and yield a much more effective system. Poor design will lead to errors, delays, and confusion. Looks different from all other icons.
- Is obvious what it does or represents. - Is recognizable when no larger than 16 pixels square.
- Looks as good in black and white as in color. Icon Size

Supply in all standard sizes.
- 16 × 16 pixels.
- 16- and 256-color versions. - 32 × 32 pixels
- 16- and 256-color versions. - 48 × 48 pixels
- 16- and 256-color versions.
- Use colors from the system palette.
- Use an odd number of pixels along each side.
- Provides center pixel around which to focus design.
- Minimum sizes for easy selection:
-With stylus or pen: 15 pixels square.
- With mouse: 20 pixels square.
- With finger: 40 pixels square. - Provide as large a hot zone as possible. Choosing Images
- Use existing icons when available.
- Use images for nouns, not verbs.
- Use traditional images.
- Consider user cultural and social norms.

The Design Process of Icons
- Define purpose:

To begin the design process, first define the icon's purpose and use. Have the design team brainstorm about possible ideas, considering real-world metaphors.
- Collect, evaluate, and sketch ideas:

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.

Course Code: CSDT124	**Course Name: Soft Computing**	**Total Lectures (48 Hours)**
Teaching Scheme : 4 hrs/week	**Examination Scheme:** **IA: 30 Marks** **UE: 70 Marks**	**No. of Credits 2**
Course Prerequisites:	• A strong mathematical background • Proficiency with algorithms • Critical thinking and problem solving skills	
Course Objectives:	• To introduce the ideas of soft computational techniques based on human experience. • To generate an ability to design, analyze and perform experiments on real life problems using various Neural Learning Algorithms. • To conceptualize fuzzy logic and its implementation for various real world applications. • To apply the process of approximate reasoning using Neuro-Fuzzy Modeling. • To provide the mathematical background to carry out optimization using genetic algorithms.	

Chapter	**Course Contents**	**No. of Lectures**
1	**Introduction to Soft Computing** Neural Networks: Definition, Advantages, Applications, Scope. Fuzzy logic: Definition, Applications. Genetic Algorithms: Definition, Applications.	2
2	**Neural Network** Fundamental Concept: Artificial Neural Network, Biological Neural Network, Brain vs. Computer-Comparison Between Biological Neuron and Artificial Neuron (Brain vs. Computer), Artificial Neurons, Neural Networks and Architectures: Neuron Abstraction, Neuron Single Functions, Mathematical Preliminaries, Neural Networks Defined, Architectures: Feedforward and Feedback, Salient Properties of NeuralNetworks 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, Non-linearly Separable Problems, Capacity of a Simple Threshold Logic Neuron, Revisiting the XOR Problem, Multilayer Networks, How Many Hidden Nodes areEnough? Learning and Memory: An Anecodatal Introduction, Long Term Memory, The Behavioral Approach to Learning, The Molecular Problem of Memory, Learning Algorithms, Error Correction and Gradient	22

	Descent Rules, Learning Objective for TLNs, Pattern Space and Weight Space. Linear Seperabilty, Hebb Network, Perceptron Network. α- Least Mean Square Learning, MSE Error Surface and Its Geometry, Steepest Descent Search with Exact Gradient Information, μ-LMS: Approximate Gradient Descent, Application of LMS TO Noise Cancelation.	
3	**Fuzzy Set Theory** Brief Review of Conventional Set Theory, Introduction to Fuzzy Sets, Properties of Fuzzy Sets, Operations on Fuzzy Sets, Crisp Relation, Fuzzy Relation, Tolerance and equivalence relation, Fuzzy Tolerance and equivalence relation, Fuzzy Max-Min and Max-Product Composition, Membership Functions, Fuzzification, Defuzzification to crisp sets, λ-Cuts for fuzzy Relations, Fuzzy (Ruled-Based) system, Graphical technique of inference, Membership value assignment-Intuition,Inference. Fuzzy Classification -Classification by equivalence relation, Cluster analysis, cluster validity, c-Means clustering, Hard c-means, Fuzzy c-Means, Fuzzy Arithmetic, Fuzzy Extension Principle	16
4	**Genetic Algorithms** What are Genetic Algorithms? Why Genetic Algorithms? Biological Background: The Cell, Chromosomes, Genetics, Reproduction, Natural Selection, Traditional Optimization and Search Techniques,Simple GA, Terminologies and Operators in GA, Encoding, Selection, Crossover, Mutation, Search Termination, Constraints in GA, Problem solving using GA, Classification of GA	8

References:

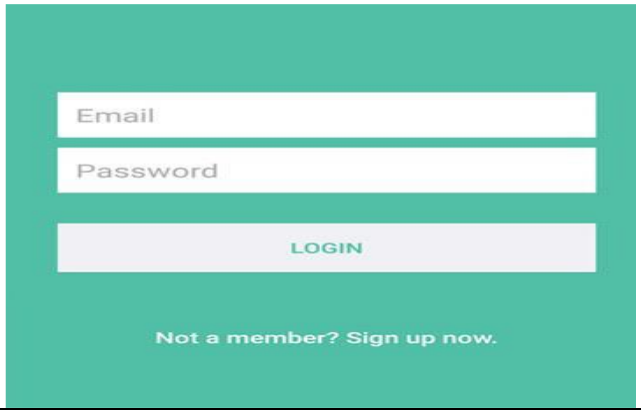
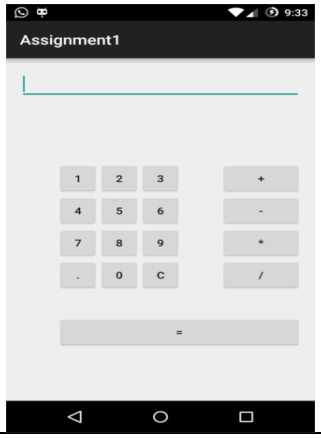
Sr. No.	Title of the Book	Author/s	Publication
1	Fuzzy Logic With Engineering Applications	Timothy Ross	Wiley Publication
2	Introduction to Soft Computing	Deepa & Shivanandan	Wiley Publication
3	Genetic Algorithms in Search, Optimization and Machine Learning	David E. Goldberg	Pearson Education
4	Fundamentals of Neural Networks – Architectures, Algorithms, And Applications	Laurene Fausett	Pearson Education
5	Neural Networks	Satish Kumar	Tata McGrawHill

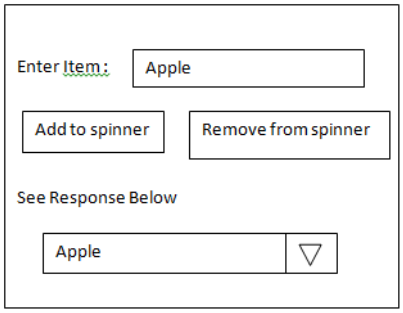
CSDP124: Soft Computing Practical Assignment

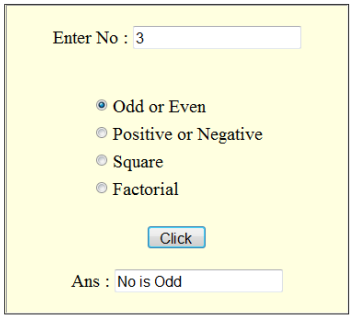
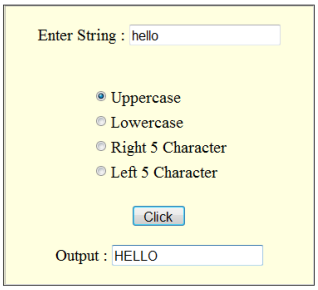
Implement the programs in C/C++/Java/MATLAB

Sr. No	Assignment
1.	Write a program to implement Fuzzy Operations Union Intersection Complement Algebraic sum Algebraic product Cartesian product
2.	Write a program to implement De Morgans law.
3.	Write a program to implement Max-Min Composition and Max-Product Composition.
4.	Write a program to implement lambda cut
5.	Write a program to implement Activation Function.
6.	Write a program to implement Perceptron Learning Rule
7.	Write a program to implement Hebb's Rule
8.	Write a program to implement Feed Forward Network
9.	Write a program for building an Artificial Neural Network by implementing the Back propagation Algorithm and test the same using appropriate data sets.
10.	Write a program for solving linearly separable problem using Perceptron Model.
11.	Write a program to develop supervised learning algorithm
12.	Write a program to study and analyze genetic life cycle

CSUP125: Practical on Advanced OS & Mobile Technologies

Sr. No.	Mobile Technologies Assignments
1.	Java Android Program to demonstrate login form with validation. 
2.	Java Android Program to demonstrate Registration form with validation.
3.	Create the simple calculator shown below also perform appropriate operation 
4.	Create an Android application which examine, that a phone number, which a user has entered is in the given format. * Area code should be one of the following: 040, 041, 050, 0400, 044 * There should 6-8 numbers in telephone number (+ area code).
5.	By using Spinner, Buttons. Write a program to draw following GUI.

	
6.	Create an Android application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many right answers were right and shows the result to user.
7.	Construct an app to display the image on date wise.
8.	Construct image switcher using setFactory().
9.	Construct a bank app to display different menu like windrow, deposite etc.
10.	Create an Android application, where the user can enter player name and points in one view and display it in another view.
11.	Create an Android application, the user can enter 10 students information and stored it in file and display student information in second view and also search the particular student information.
12.	Write an application to accept two numbers from the user, and displays them, but reject input if both numbers are greater than 10 and asks for two new numbers.
13.	Create table Customer (id, name, address, phno). Create Application for Performing the following operation on the table. (using sqlite database) i) Insert New Customer Details. ii) Show All the Customer Details
14.	Create an application that allows the user to enter a number in the textbox named 'getnum'. Check whether the number in the textbox 'getnum' is palindrome or not. Print the message accordingly in the label control named lbldisplay when the user clicks on the button 'check'.
15.	Create Following Table: Emp (emp_no,emp_name,address,phone,salary) Dept (dept_no,dept_name,location) Emp-Dept is related with one-many relationship. Create application for performing the following Operation on the table 1) Add Records into Emp and Dept table. 2) Accept Department name from User and delete

	employee information which belongs to that department.
16.	Perform following numeric operation according to user selection of radio button 
17.	Perform following string operation according to user selection of radio button. 
18.	Java Andorid Program to Perform all arithmetic Operations using Calculators
19.	Java Android Program to Change the Image Displayed on the Screen
20.	Java Android Program to Demonstrate Alert Dialog Box
21.	Java Android Program to Demonstrate the Menu Application
22.	Java Android Program to Demonstrate List View Activity with all operations (Insert, delete, Search).
23.	Java Android Program to Display SMS from the Phone Numbers, which are in Your Contacts
24.	Java Android Program to send email with attachment.
25.	Create an Android application which will ask the user to input his name and a message, display the two items concatenated in a label, and change the format of the label using radio buttons and check boxes for selection, the user can make the label text bold, underlined or italic and change its color .include buttons to display the message in the label, clear the text boxes and label and then exit.
26.	Write a program to search a specific location on Google Map.
27.	Write a program to perform Zoom In, Zoom Out operation and display Satellite view,Terrain view of

	current location on Google Map.
28.	Digital Bio Data PhoneGap Application using HTML5.
29.	Write a PhoneGap application to display push notification.
30.	Write a PhoneGap application to create a contact, Searching for Contacts, Cloning Contacts, Removing Contacts.
31.	Write a IOS application to display “Hello World’.
32.	Write aios application to display gesture recognizer.
33.	Write a Swift program to add the last character (given string) at the front and back of a given string. The length of the given string must be 1 or more.
34.	Write a Swift program to create a new string where all the character "a" have been removed except the first and last positions.
35.	Write a Swift program to create a new string made of 2 copies of the first 2 characters of a given string. The string may be any length.
36.	Students design mobile applications for the Android or iOS platforms that uniquely meet clear needs in today’s markets. Student design documents include narratives, categorized use cases, screen rows, and database schemata
37.	Handling button events / actions in iOS
38.	Handling image in iOS using ImageView
39.	Write a iOS application to implement UI elements like ScrollView, TableView, Pickers, Switches
40.	Write a iOS application to Managing camera in iOS
41.	Write a iOS application to Handling audio, video and file in iOS
42.	Write a iOS application to Handling Accelerometer to manage change in position