M. C. E. Society's

Abeda Inamdar Senior College

Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)

Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

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

Objectives of Course:

Sr. No.	Objective
1.	To provide advanced and in-depth knowledge of computer science and its applications.
2.	To prepare Post Graduates who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, design and implementation skills.
3.	To enable students pursue a professional career in Information and Communication Technology in related industry, business and research.
4.	To impact professional knowledge and practical skills to the students.

Program specific Outcome

Sr. No.	Objective
1.	Provides technology-oriented students with the knowledge and ability to develop creative solutions
2.	Develop skills to learn new technology.
3.	Apply computer science theory and software development concepts to construct computing-based solutions.
4.	Design and develop computer programs/computer-based systems in the areas related to algorithms, networking, web design, cloud computing, Artificial Intelligence, Mobile applications.
5.	An understanding of professional, ethical, legal, security, and social issues and responsibilities for the computing profession.

M. C. E. Society's

Abeda Inamdar Senior College

Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

M. Sc. Computer Science-I

(CBCS – Autonomy 21 Pattern)

Course/ Paper Title	**Paradigm of Programming Language**
Course Code	21SMCS111
Semester	I
No. of Credits	4

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To Prepare student to think about programming languages analytically
2.	Separate syntax from semantics
3.	Compare programming language designs, understand their strengths and weaknesses
4.	Learn new languages more quickly
5.	Understand basic language implementation techniques
6.	Learn small programs in different programming Languages

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Students will acquire thinking of different programming language.
2.	Students will become aware of basic language implementation techniques.
3.	Students will understand the Significance of learning new programming language.

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Introduction**	**04**
	1. The Art of Language Design 2. The Programming Language Spectrum 3. Why Study Programming Languages? 4. Compilation and Interpretation 5. Programming Environments	02 02
Unit II	**Names ,Scopes ,and Bindings**	**08**
	1. The Notion of Binding Time 2. Object Lifetime and Storage Management 3. Static Allocation, Stack-Based Allocation,Heap-Based Allocation, Garbage CollectionScopeRules 4. Static Scoping, Nested Subroutines,Declaration Order, Dynamic Scoping The meaning of Names in a Scope 5. Aliases, Overloading, Polymorphism and Related Concepts, the Binding of Referencing Environments 6. Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures MacroExpansion	01 01 02 02 02
Unit III	**Control Flow**	**05**
	1. Expression Evaluation , Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation 2. Structured and Unstructured Flow, Structured Alternatives to go-to sequencing 3. Selection - Short-Circuited Conditions, Case/Switch Statements Iteration 4. Iteration- Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion 5. Recursion- Iteration and Recursion, Applicative-and Normal-Order Evaluation	02 02 01
Unit IV	**Data Types**	**10**
	1. Introduction 2. Primitive Data Types 3. Numeric Types : Integer, Floating point, Complex, Decimal,	02

	Boolean Types, Character Types 4. Character String Types 5. Design Issues, Strings and Their Operations, String Length Operations, Evaluation, Implementation of Character String Types 6. User defined Ordinal types Enumeration types,Designs Evaluation Sub range types, Ada'sdesign Evaluation Implementation of user defined ordinal types 7. Array types 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 9. Associative Arrays 10. Structure and operations, Implementing associative arrays, 11. Record types 12. Definitions of records, References to record fields, Operations on records, Evaluation, Implementation of Record types 13. Union Types 14. Design issues, Discriminated versus Freeunions, Evaluation, Implementation of Union types 15. Pointer and Reference Types 16. Design issues, Pointer operations, Pointer problems, Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation 17. Implementation of pointer and reference types 18. Representation of pointers and references Solution to dangling pointer problem Heap management	02 02 02 02
Unit V	**Subprograms and Implementing Subprograms**	**05**
	1. Introduction 2. Fundamentals of Subprograms 3. Design Issues for subprograms 4. Local Referencing Environments 5. Parameter-Passing Methods 6. Parameters That Are Subprograms 7. Overloaded Subprograms 8. Generic Subroutines, Generic Functions in C++ 9. Design Issues for Functions 10. User-Defined Overloaded Operators 11. Coroutines 12. Implementing Subprograms 13. The General Semantics of Calls and Returns 14. Implementing "Simple" Subprograms 15. Implementing Subprograms with Stack-Dynamic Local Variables	02 03

	16. Nested Subprograms 17. Blocks 18. Implementing Dynamic Scoping	
Unit VI	**Data Abstraction and Object Orientation**	**08**
	1. Object-Oriented Programming 2. Encapsulation and Inheritance Modules, Classes, Nesting(Inner Classes), Type Extensions, Extending without Inheritance 3. Initialization and Finalization Choosing a Constructor, References and Values, Execution Order, Garbage Collection 4. Dynamic Method Binding 5. Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures 6. Multiple Inheritance 7. Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-Inheritance	01 02 03 01 01
Unit VII	**Concurrency**	**05**
	1. Introduction : Multiprocessor Architecture Categories of concurrency, Motivations for studying concurrency 2. Introduction to Subprogram-level, concurrency Fundamental concepts, Language Design for concurrency, Design Issues 3. Semaphores - Introduction Cooperation synchronization, Competition Synchronization ,Evaluation	02 02 01
Unit VIII	**Functional Programming in Scala**	**15**
	1. Introduction to Scala 2. Scala Data type 3. Scala variables 4. Scala operators and ControlStructures 5. Scala Classes and objects 6. Scala Function 7. Array 8. Scala Collection(List,Set,Map) 9. Scala as FunctionalProgramming i. Function call by name ii. Anonymous Function iii. Higher order function	05 05 05

References:

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/ Paper Title	**Design and Analysis of Algorithm**
Course Code	21SMCS112
Semester	I
No. of Credits	4

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To prepare students to design the algorithms analysis of algorithms
2.	To select the appropriate algorithm by doing necessary
3.	To learn basic Algorithm Analysis techniques and understand the use of asymptotic notation
4.	Understand different design strategies
5.	Understand the use of data structures in improving algorithm performance
6.	Understand classical problem and solutions and classification of problems
7.	To provide foundation in algorithm design and analysis
8.	To develop ability to understand and design algorithms in context of space and time complexity.

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Students will know different design strategies of designing an algorithm
2.	Students will understand how data structure will be used in improving performances of an algorithm
3.	Students will understand the calculation of time and space complexity.
4.	Students will aware of classical problem and solutions and classification of problems

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Basics of Algorithms**	**06**
	1. Algorithm definition and characteristics 2. Space complexity 3. Time complexity, worst case-bestcase-average case 4. complexity, asymptotic notation 5. Recursive and non-recursive algorithms	02 02 02
Unit II	**Divide and conquer strategy**	**08**
	1. General method, control abstraction 2. Binary search 3. Merge sort, Quicksort 4. Comparison between Traditional Method of Matrix Multiplication vs. Strassen's Matrix Multiplication	02 02 02 02
Unit III	**Greedy Method**	**08**
	1. Knapsack problem 2. Job sequencing with deadlines, 3. Minimum-cost spanning trees: Kruskal and Prim's algorithm 4. Optimal storage on tapes 5. Optimal merge patterns 6. Huffman coding 7. Shortest Path :Dijkstra's Algorithm	02 02 02 02
Unit IV	**Dynamic Programming**	**10**
	1. Principle of optimality 2. Matrix chain multiplication 3. 0/1 Knapsack Problem i. Merge & Purge ii. Functional Method 4. Bellman Ford Algorithm 5. All pairs Shortest Path Floyd	02 04 02

	Warshall Algorithm 6. Longest common subsequence, 7. String editing, Travelling Sales person problem	02
Unit V	**Decrease and Conquer**	**8**
	1. Definition of Graph, Representation of Graph 2. By Constant - DFS and BFS 3. Topological sorting 4. Connected components and spanning trees 5. By Variable Size decrease Euclid's algorithm 6. Articulation Point and Bridges	01 02 01 01 02 01
Unit VI	**Backtracking**	**8**
	1. General method 2. Fixed Tuple vs. Variable Tuple Formulation 3. n- Queen's problem 4. Graph coloring problem 5. Hamiltonian cycle 6. Sum of subsets	01 02 02 01 01 01
Unit VII	**Branch and Bound**	**8**
	1. Introduction 2. FIFO BB Search, LIFO Search 3. Definitions of LCBB Search 4. Bounding Function, Ranking Function 5. Traveling Salesman problem Using Variable tuple 6. Formulation using LCBB 7. 0/1 knapsack problem using LCBB	01 02 01 02 01 01
Unit VIII	**Problem Classification**	**4**
	1. Non deterministic algorithm 2. The class of P, NP, NP-hard and NP - Complete problems	02 02

References:

1. Computer algorithms, Ellis Horowitz, Sartaj Sahni & Sangu the var Rajasekaran, Galgotia Publication
2. Algorithms , T. Cormen, C. Leiserson, & R. Rivest, , MIT Press
3. The Design and Analysis of Computer Algorithms, A. Aho, J. Hopcroft& J. Ullman, Addison Wesley
4. The Art of Computer Programming, Donald Knuth, Addison Wesley
5. The Algorithm Manual, Steven Skiena, Springer
6. Graphs, Networks and Algorithms, Jungnickel, Springer

Course/ Paper Title	**NoSQL Database Technologies**
Course Code	21SMCS113
Semester	I
No. of Credits	4

Aims & Objectives of the Course

Sr. No.	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.

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Student will know almost all concepts of NoSQL
2.	Student will able to compare various types of NoSQL databases.
3.	Student will able to decide what database technology to use for particular application.

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Introduction to NOSQL (Core concepts)**	**20**
	1. Why NoSQL **2.** Aggregate Data Models **3.** Data modeling details **4.** Distribution Models **5.** Consistency **6.** Version stamps **7.** Map-Reduce	
Unit II	**Implementation with NOSQL databases**	**16**
	1. Key-Value Databases (Risk) 2. Document Databases (Mongodb) 3. Column-Family stores(Cassandra) 4. Graph databases (Neo4j)	
Unit III	**Schema Migrations**	**7**
Unit IV	**Poly got Persistence (Multi model types)**	**7**
Unit V	**Beyond NoSQL**	**5**
Unit VI	**Choosing your database**	**5**

References:

1. NoSQL Distilled, Pramod Sadalge, Martin Fowler
2. NoSQL for Dummies, A Willy Brand
3. http://nosql-database.org

Course/ Paper Title	**Cloud Computing**
Course Code	**21SMCS114A**
Semester	I
No. of Credits	2

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To understand the principles and paradigm of Cloud Computing
2.	To appreciate the role of Virtualization Technologies
3.	Ability to design and deploy Cloud Infrastructure
4.	Understand cloud security issues and solutions

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Student will know about basics of Cloud Computing
2.	Student will be aware of role of Virtualization Technologies
3.	Student will understand security issues and solutions

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Introduction to Cloud Computing**	**8**
	1. Overview, Layers and Types of Cloud, i. Desired Features of a Cloud ii. Benefits and Disadvantages of Cloud Computing iii. Cloud Infrastructure Management iv. Infrastructure as a Service Providers v. Platform as a Service Providers vi. Multitenant Technology.	

	2. Cloud-Enabling Technology: i. Broadband Networks and Internet Architecture ii. Data Center Technology, iii. Virtualization Technology. 3. Infrastructure as a Service 4. Platform as a Service 5. Software as a Service 6. Cloud Deployment Models.	
Unit II	**Abstraction and Virtualization**	**7**
	1. Introduction to Virtualization Technologies 2. Load Balancing and Virtualization 3. Understanding Hyper visors 4. Virtual Machines Provisioning and Manageability Virtual Machine Migration Services 5. Provisioning in the Cloud Context Virtualization of CPU, Memory , I/O Devices, Virtual Clusters and Resource management	
Unit III	**Programming, Environments and Applications**	**8**
	1. Features of Cloud and Grid Platforms 2. Programming Support of Google App Engine 3. Programming on Amazon AWS and Microsoft Azure 4. Emerging Cloud Software Environments, 5. Applications: i. Moving application tocloud, ii. Microsoft Cloud Services- introduction to SLA and Cloud pricing, iii. Google Cloud Applications, iv. Amazon Cloud Services, v. Cloud Applications.	
Unit IV	**Security In The Cloud**	**7**
	1. Security Overview – Cloud Security Challenges and Risks 2. Software-as-a-Service Security – Security Governance 3. Risk Management – Security Monitoring 4. Security Architecture Design i. Data Security ii. Application Security iii. Virtual Machine Security 5. Identity Management and Access Control 6. Disaster Recovery in Clouds.	

References:

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. Distributed and Cloud Computing, From Parallel Processing to the Internet of Things, Kai Hwang, Geoffrey C Fox, Jack G Dongarra, Morgan Kaufmann Publishers, 2012.

Course/ Paper Title	**Cloud Computing Practical**
Course Code	**21SMCS115A**
Semester	I
No. of Credits	2

Aims & Objectives of the Course

Sr. No.	**Objectives**
1.	To understand basic cloud computing concepts.
2.	To understand implementation of AWS
3.	To understand implementation of MS-Azure

Expected Course Specific Learning Outcome

Sr. No.	**Learning Outcome**
1.	Students will know basics of cloud computing.
2.	Students will able to implement cloud using AWS
3.	Students will able to implement cloud using MS-Azure

Syllabus

Unit No	Title with Contents	No. of Practical Sessions
Unit I	Working and Implementation of Infrastructure as a service. (AWS)	15
	Working and Implementation of Software as a service	
	Working and Implementation of Platform as a services. (Azure)	
	Practical Implementation of Storage as a Service using GCP provider	
	Working of Google drive to make spreadsheet and notes.	
	Working and Implementation of identity management.	
	Write a program for web feed.	
	Execute the step to Demonstrate and implementation of cloud on single sign on.	
	Practical Implementation of cloud security.	
	Installing and Developing Application Using Google App Engine.	
	Implement VM WARE ESXi Server	
	Using Open Nebula to manage heterogeneous distributed data center Infrastructure.	
	Implementation of Cloud Failure Cluster.	
	Managing and working of cloud Xen server.	
	Working with Aneka and demonstrate how to Managing cloud computing Resources.	
	Installation and configuration of cloud Hadoop and demonstrate simple query.	
	Create a sample mobile application using Amazon Web Service (AWS) account as a cloud service. Also provide database connectivity with implemented mobile application.	

Course/ Paper Title	**Artificial Intelligence**
Course Code	**21SMCS114B**
Semester	I
No. of Credits	2

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To learn various types of algorithms useful in Artificial Intelligence (AI).
2.	To convey the ideas in AI research and programming language related to emerging technology.
3.	To understand the numerous applications and huge possibilities

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Student will learn various types of algorithms useful in Artificial Intelligence (AI).
2.	Student will be aware of Python programming language.
3.	Student will understand the numerous applications of Artificial Intelligence (AI).

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Introduction to Artificial Intelligence**	**2**
	1.Introduction and Intelligent systems, 2. What Is AI, The Foundations of Artificial Intelligence, 3. The History of Artificial Intelligence, Applications of AI, Early work in AI and related fields.	1 1
Unit II	**Searching:**	**8**
	1. Defining AI problems as a State Space Search: example, Search and Control Strategies, Problem Characteristics, Issues in Design of Search Programs, Production System. **2. Blind Search Techniques :** i. BFS ii. DFS iii. DLS iv. Iterative Deepening Search v. Bidirectional Search vi. Uniform cost Search. **3. Heuristic search techniques:** i. Generate and test ii. Hill Climbing iii. Best First search iv. Constraint Satisfaction v. Mean-End Analysis vi. A* vii. AO*.	**1** **3** 4
Unit III	**Knowledge Representation:**	**8**
	1. Representations and Mappings, 2. Approaches to Knowledge Representation i. Knowledge representation method ii. Propositional Logic iii. Predicate logic iv. Representing Simple facts in Logic v. Resolution, 3. Game Playing i. Minimax Search Procedures ii. Adding alpha-beta cutoffs.	1 5 2

Unit IV	**Agent, Environment and Introduction to Expert System**	**6**
	1. What are Agent and Environment, i. Structure and types of Agents, ii. nature of Environment, iii. Properties of Environment 2. What are Expert System, i. Components of Expert System ii. Forward and backward chaining iii. Applications of Expert System,	3 3
Unit V	**Introduction to AI with Python**	**6**
	1. Introduction to Python 2. why python with AI 3. Features of Python 4. Basics of Python i. Python statements ii. Methods & Functions using python iii. Basic and advanced modules &Packages iv. Python Decorators and generators v. Advanced Objects & Data structures.	2 1 3

References:

1. Computational Intelligence, Eberhart, Elsevier Publication
2. Artificial Intelligence: A New Synthesis, Nilsson, Elsevier Publication
3. Artificial Intelligence with Python, Prateek Joshi, 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

Course/ Paper Title	**Artificial Intelligence Practical**
Course Code	**21SMCS115B**
Semester	I
No. of Credits	2

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To understand basic artificial intelligence.
2.	To understand implementation of artificial intelligence
3.	To study various methods for solving particular problem

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Students will know basics of artificial intelligence.
2.	Students will able to implement solution to real word problem.
3.	Students will able to select appropriate strategy for particular problem.

Syllabus

Unit No	Title with Contents	No. of Practical Sessions
Unit I	Program to print multiplication table for given no.	15
	Program to check whether the given no is prime or not.	
	Program to find factorial of the given no	
	program to check whether the given year is Leap or Not	
	Python program to print Fibonacci series up to n^{th} Terms.	
	Write a menu driven program in Python to perform following operations: a) add b)subtract c)multiply d)division	
	Write a program to implement List Operations (Nested list, Length, Concatenation, Membership ,Iteration ,Indexing and Slicing),	
	Write a program to implement List Methods(Add, Append, Extend & Delete)	
	Write a program to implement map, reduce and filter function with lambda function in python	
	Write a program to Illustrate Different Set Operations.	
	Write a program to implement Simple Chat bot.	
	Write a program to implement Breadth First Search Traversal.	
	Write a program to implement Depth First Search Traversal.	
	Write a program to implement Water Jug Problem.	

Course/ Paper Title	**Web Services**
Course Code	**21SMCS114C**
Semester	I
No. of Credits	2

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To understand 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 understand the concept of Restful system.

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Student will know about various web services technologies.
2.	Student will able to deploy client and server web service.
3.	Student will know about Restful system.

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Web Service and SOA fundamentals Introduction to Web Services**	**6**
	1. The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services. 2. Web Services Architecture **i.** Web services Architecture and its characteristics.	2 4

	ii. core building blocks of web services **iii.** Standards and technologies available for implementing web services **iv.** Web services communication models, **v.** Basic steps of implementing web services.	
Unit II	**SOAP: Simple Object Access Protocol**	**8**
	1. Inter-application communication and wire protocols 2. SOAP as a messaging protocol 3. Structure of a SOAP message 4. SOAP communication model 5. SOAP Web Services i. Building SOAP Web Services ii. Developing SOAP Web Services using Java, iii. Error handling in SOAP iv. Advantages and disadvantages of SOAP.	**1** 2 1 4
Unit III	**Describing and Discovering Web Services**	**8**
	1. WSDL i. WSDL in the world of Web Services, ii. Web Services life cycle iii. anatomy of WSDL definition document iv. WSDL bindings v. WSDL Tools vi. limitations of WSDL vii. Service discovery viii. role of service discovery in a SOA ix. service discovery mechanisms **2. UDDI** i. UDDI Registries ii. uses of UDDI Registry iii. Programming with UDDI iv. UDDI data structures v. Support for categorization in UDDI Registries vi. Publishing API, Publishing information to a UDDI Registry, vii. searching information in a UDDI Registry, viii. deleting information in a UDDI Registry, limitations of UDDI.	4 4
Unit IV	**The REST Architectural style**	**8**
	1. Introducing HTTP 2. The core architectural elements of a Restful system, i. Description and discovery of Restful web services, ii. Java tools and frameworks for building Restful web	2 6

	services, iii. JSON message format and tools and frameworks around JSON, iv. Build Restful web services with JAX-RS APIs, v. The Description and Discovery of Restful Web Services, vi. Design guide lines for building Restful web services vii. Secure Restful web services	

References:

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

Course/ Paper Title	**Web Services Practical**
Course Code	**21SMCS115C**
Semester	I
No. of Credits	2

Aims & Objectives of the Course

Sr. No.	**Objectives**
1.	To understand basic concepts of web services
2.	To understand how to develop web services using Java/PHP/.Net
3.	To study various methods to implement web services.

Expected Course Specific Learning Outcome

Sr. No.	**Learning Outcome**
1.	Students will know basics of web services.
2.	Students will able to implement web services.

3.	Students will able to select appropriate strategy implementing web services as per need of problem.

Syllabus

Unit No	Title with Contents	No. of Practical Sessions
Unit-I	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.	15
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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.	
	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	

Course/ Paper Title	**PPL and NoSQL Database Technologies Practical**
Course Code	**21SMCS116**
Semester	I
No. of Credits	4

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To understand basic concepts of PPL and NoSQL Database Technologies
2.	To understand how to develop Neo4j database
3.	To understand structure of MongoDB

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Students will know basics PPL and NoSQL Database Technologies
2.	Students will able to develop database using Neo4j.
3.	Students will able to use MongoDB for developing solution to particular problem.

Syllabus

Unit No	Title with Contents	No. of Practical Sessions
	LIST OF SCALA PROGRAMS (PPL)	15
UNIT I	**Control Structures** 1. Write a program to calculate average of all numbers between n1 and n2(eg.100 to 300Read 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.	

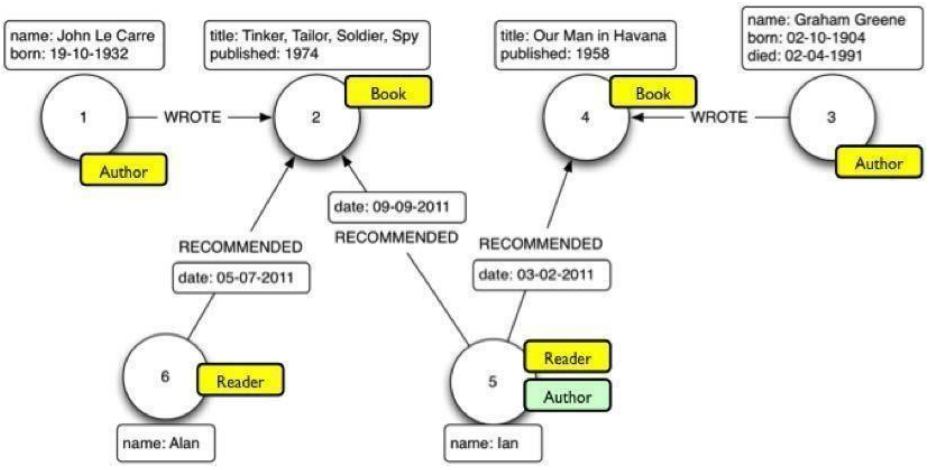
	4. Write a program to find second maximum number of four given numbers. 5. Write a program to calculate sum of prime numbers between 1 to100 6. Write a program to read an integer from user and convert it to binary and octal using user defined functions	
UNIT II	**Arrays** 1. Write a program to find maximum 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)	
UNIT III	**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.	
UNIT IV	Classes and Objects 1. Define a class Current Account (accNo, name, balance, minBalance). Define appropriate constructors and operations withdraw(), deposit(), view Balance(). 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 Purchase Order & Sales Order with members Vendor and Customer. Create object of each Purchase Order and Sales Order. 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.	
UNIT V	**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 to50. 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 of10. 7. Create a list of 50 members using function 2n+3. Create second list excluding all elements multiple of7.	
UNIT VI	**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 Roll no and First Name. Print all student information with same First Name. **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	
	NoSQL Database Technologies Practical	15
UNIT I	**MongoDB Practical Assignment 1** 1. Create a database with the name 'Movie'. 2. A 'Film' is a collection of documents with the following fields: a. Film Id b. Title of the film c. Year of release d. Genre / Category (like adventure, action, sci-fi, romantic etc.) A film can belong to more than one genre. e. Actors (First name and Last name) A film can have more than one actor. f. Director (First name and Last name) A film can have more than one director. g. Release details (It consists of places of release, dates of release and rating of the film.) 3. An 'Actor' is a collection of documents with the following fields: a. Actor Id b. First name c. Last Name d. Address (Street, City, State, Country, Pin-code) e. Contact Details (Email Id and Phone No) f. Age of an actor. **Queries:** 1. Insert at least 10 documents in the collection Film– a. Insert at least one document with film belonging to two genres. b. Insert at least one document with film that is released at more than one place and on two different dates. c. Insert at least three documents with the films released in the same year. d. Insert at least two documents with the films directed by	

	one director. Insert at least two documents with films those are acted by a pair 'Madhuri Dixit' and 'Shahrukh Khan' 2. Insert at least 10 documents in the collection Actor. Make sure, you are inserting the names of actors who have acted in films, given in the 'Film' collection. 3. Display all the documents inserted in both the collections. 4. Add a value to the rating of the film whose title starts with 'T'. 5. Add an actor named"_____" in the 'Actor' collection. Also add the details of the film in 'Film' collection in which this actor has acted in. 6. Delete the film"_____". 7. Delete an actor named"_____". 8. Delete all actors from an 'Actor' collection who have age greater than" 9. Update the actor's address where Actor Id is "". 10. Update the genre of the film directed by"	
UNIT II	**MongoDB Practical Assignment 2** 1. Create a database with name 'Company'. 2. An 'Employee' is a collection of documents with the following fields: a. Employee ID b. First Name c. Last Name d. Email e. PhoneNo. f. Address (House No, Street, City, State, Country, Pin-code) g. Salary h. Designation i. Experience j. Date of Joining k. Birthdate 3. A 'Transaction' is a collection of documents with the following fields: a. Transaction Id, b. Transaction Date	

	c. Name (First Name of employee who processed the transaction) d. Transaction Details (Item Id, Item Name, Quantity, Price) e. Payment (Type of Payment (Debit/Credit/Cash), Total amount paid, Payment Successful) f. Remark (Remark field can be empty.) **Queries:** 1. Insert at least 5 documents in 'Employee' collection. 2. Insert multiple documents (at least 10) into the 'Transaction' collection by passing an array of documents to the db. Collection insert () method. 3. Display all the documents of both the collections in a formatted manner. 4. Update salary of all employees by giving an increment of Rs.4000. 5. Update the remark for transaction id201. 6. Update designation of an employee named"_ " from supervisor to manager. 7. Update designation of an employee having Employee Id as 8. Change the address of an employee having Employee Id as 9. Delete transaction made by"_____" employee on the given date. 10.Delete all the employees whose first name starts with 'K'.	
UNIT III	**MongoDB Practical Assignment 3** 1. Find the titles of all the films starting with the letter 'R' released during the year 2009 and2011. 2. Find the list of films acted by an actor"_____". 3. Find all the films released in90s. 4. Find all films belonging to "Adventure" and "Thriller" genre. 5. Find all the films having 'A' rating. 6. Arrange the film names in ascending order and release year should be in descending order. 7. Sort the actors in ascending order according to their age. 8. Find movies that are comedies or dramas and are released after2013. 9. Show the latest 2 films acted by an actor"_____". 10. List the titles of films acted by actors "_____"and"____ 12. Retrieve films with actor details.	

UNIT IV	**MongoDB Practical Assignment 4** 1. Find employees having designation as either ‘manager’ or ‘floor supervisor’. 2. Find an employee whose name ends with"_____ 3. Display the name of an employee whose salary is greater than_____ 4. Sort the employees in the descending order of their designation. 5. Count the total number of employees in a collection. 6. Calculate the sum of total amount paid for all the transaction documents. 7. Calculate the sum of total amount paid for each payment type. 8. Find the transaction id of the latest transaction. 9. Find designation of employees who have made transaction of amount greater than Rs. 500. 10. Find the total quantity of a particular item sold using Map Reduce.	
UNIT V	**Neo4j Practical Assignment 1** Create the following databases as graph models. Visualize the models after creation, Return properties of nodes, Return the nodes labels, Return the relationships with its properties. **NB:** You may assume and add more labels , relationships, properties to the graphs **1.** Create a library database , as given below. 	

There are individual books, readers, and authors that are present in the library data model.. A minimal set of labels are as follows:
Book: This label includes all the books
Person: This label includes authors, translators, reviewers, Readers, Suppliers and so on
Publisher: This label includes the publishers of books in the database

A set of basic relationships are as follows:
Published By: This relationship is used to specify that a book was published by a publisher **Votes**: This relationship describes the relation between a user and a book, for example, how a book was rated by a user.
Reviewed By: This relationship is used to specify that a book was reviewed and remarked by a user.
Translated By: This relationship is used to specify that a book was translated to a language by a user.
Issued By: This relationship is used to specify that a book was issued by a user. Returned By: This relationship is used to specify that a book was returned by a user

Every book has the following properties:
Title: This is the title of the book in string format

Tags: This is an array of string tags useful for searching through the database based on topic, arguments, geographic regions, languages, and so on
Status: the book status, specifying whether its issued or in library.
Condition: book condition, new or old
Cost : Cost of book
Type: book is a Novel, Journal, suspense thriller etc

2. Consider a Song database, with labels as Artists, Song, Recording_company, Recoding_studio, song author etc.
Relationships can be as follows
Artist ⟶ [Performs] ⟶ Song ⟶ [Written by] ⟶ Song_author.
Song ⟶ [Recorded in] ⟶ Recording Studio ⟶ [managed by] ⟶ recording Company Recording Company ⟶ [Finances] ⟶ Song
You may add more labels and relationship and their properties, as per assumptions.

3. Consider an Employee database, with a minimal

	set of labels as follows Employee: denotes a person as an employee of the organization Department: denotes the different departments, in which employees work. Skillset: A list of skills acquired by an employee Projects: A list of projects in which an employee works. A minimal set of relationships can be as follows: Works_in : employee works in a department Has_acquired: employee has acquired a skill Assigned_to : employee assigned to a project Controlled_by: A project is controlled by a department Project_manager : Employee is a project_manager of a Project **4.** Consider a movie database, with nodes as Actors, Movies, Roles, Producer, Financier, Director. Assume appropriate relationships between the nodes, include properties for nodes and relationships. **5.** Create a Social network database, with labels as Person, Affiliations, Groups, Story, Timeline etc. Some of the relationships can be as follows: Person ⟶ [friend of] ⟶ Person ⟶ [affiliated to] ⟶ affiliations Person ⟶ [belongs to] ⟶ Groups,Person ⟶ [create] ⟶ Story ⟶ [refers to] ⟶ Person Person ⟶ [creates] ⟶ Timeline ⟶ [referencefor] ⟶ Story, Timeline ⟶ [contains] ⟶ Messages	
UNIT VI	**Neo4j Practical Assignment 2 Simple Queries**. 1. Library Database: a) List all people, who have issued a book"……" b) Count the number of people who have read "…." c) Add a property "Number of books issued " for Mr. Joshi and set its value as the count d) List the names of publishers from pune city.	

	2. Song Database: a) List the names of songs written by":….." b) List the names of record companies who have financed for the song"…." c) List the names of artist performing the song"….." d) Name the songs recorded by the studio "……" 3. Employee Database: a) List the names of employees in department" ....." b) List the projects along with their properties, controlled by department"……" c) List the departments along with the count of employees init d) List the skill set for an employee"..... " 4. Movie Database: a) Find all actors who have acted in a movie"....." b) Find all reviewer pairs, one following the other and both reviewing the same movie, and return entire sub graphs. c) Find all actors that acted in a movie together after 2010 and return the actor names and movie node d) Find all movies produced by " " 5. Social Network Database: a) Find all friends of "John", along with the year, since when john knows them. b) List out the affiliations of John. c) Find all friends of john, who are born in the same year as John d) List out the messages posted by John in his timeline, during the year2015.	
UNIT VII	**Neo4j Assignment 3 Complex pattern Queries:** 1. Library database a) List all readers who have recommended either book "…" or "……."or "……. b) List the readers who haven't recommended any book c) List the authors who have written a book that has been read / issued by maximum number of readers. d) List the names of books recommended by"....." And	

read by at least one reader

e) List the names of books recommended by“..... ” and read by maximum number of readers.

f) List the names of publishers who haven’t published any books written by authors from Pune and Mumbai.

g) List the names of voracious readers in our library

2. Song Database:

a) List the names of artists who have sung only songs written by“.....

b) List the names of artists who have sung the maximum number of songs recorded by“……” studio

c) List the names of songs financed by “…..”, and sung by“ .....

3. Employee Database:

a) List the names of employees having the same skills as employee“ .....

b) List the projects controlled by a department “……..... ” and have employees of the same department working in it.

c) List the names of the projects belonging to departments managed by employee“.....

4. Movie Database:

a) List the names of actors that paired in multiple movies together.

b) List all pairs of actor–movie sub graphs along with the roles played.

c) List all reviewers and the ones they are following directly or via another a third Reviewer

d) List the names of movies that have the most number of reviews.

5. Social Network Database:

a) List out the people, who have created maximum time line messages.

b) List all friends of John’s friend, Tom

c) List the people with maximum friends

d) List the people who are part of more than 3groups.

Course/ Paper Title	**Advanced Operating System**
Course Code	**21SMCS121**
Semester	II
No. of Credits	4

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To understand the programming interface to the Unix/Linux system – the system call interface.
2.	To understand the functions of Operating Systems.
3.	To get an insight into functional modules of Operating Systems.
4.	To understand the concepts underlying in the design and implementation of Operating Systems.

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Student will able to implement various system call interfaces.
2.	Student will able to design functional modules of operating system.
3.	Student will able to use systems calls for implementing various functions in programs.

Syllabus

Unit No	Title with Contents	No. of Lectures
Unit I	**Introduction to UNIX/Linux Kernel**	**06**
	1. System Structure 2. User Perspective 3. Assumptions about Hardware 4. Architecture of UNIX Operating System (TextBook-1:ChapterTopics:1.2,1.3,1.5,2.1) 5. Concepts of Linux Programming i. Files and the File system ii. Processes iii. Users and Groups iv. Permissions v. Signals vi. Inter process Communication (TextBook-3: Chapter 1- relevant topics)	**01** **01** **02** **02**
Unit II	**File and Directory I/O**	**18**
	1. Buffer headers 2. Structure of the buffer pool 3. Scenarios for retrieval of a buffer 4. Reading and writing disk blocks 5. Inodes 6. Structure of regular file i. Open ii. Read iii. Write iv. Lseek v. Close vi. Pipes vii. 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) viii. creat ix. file sharing x. atomic operations xi. dup2 xii. sync xiii. fsync and fdatasync xiv. fcntl xv. /dev/fd xvi. stat, fstat, lstat	**02** **02** **02** **02** **10**

	xvii. file types xviii. Set-User-ID and Set-Group-ID xix. file access permissions xx. ownership of new files and directories xxi. access function xxii. umask function xxiii. chmod and fchmod xxiv. sticky bit xxv. chown, fchown, and lchown xxvi. file size xxvii. file truncation xxviii. file systems xxix. link, unlink, remove, and rename functions xxx. symbolic links xxxi. symlink and readlink functions xxxii. file times, utime xxxiii. mkdir and rmdir xxxiv. reading directories xxxv. chdir, fchdir, and getcwd xxxvi. device special files (TextBook-2: Chapter Topics: 3.3, 3.4, 3.10-3.14, 3.16, 4.2-4.23)	
Unit III	**Process Environment, Process Control and Process Relationships**	**18**
	1. Process states and transitions 2. layout of system memory 3. the context of a process i. saving the context of a process ii. sleep iii. process creation iv. signals v. process termination vi. awaiting process termination vii. invoking other programs viii. the user id of a process ix. changing the size of the process 4. The Shell, Process Scheduling (TextBook-1: Chapter Topics: 6.1-6.4, 6.6, 7.1-7.8,8.1) 5. Process termination 6. environment list 7. memory layout of a C program i. shared libraries ii. environment variables iii. setjmp and longjmp iv. getrlimit and setrlimit v. process identifiers vi.	**02** **04** **02** **02** **04**

	a. Fork b. Vfork c. Exit d. wait and waitpid e. waited f. wait3 and wait4 g. race conditions h. exec 8. changing user IDs and group IDs 9. system function 10. user identification 11. 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)	**02** **02**
Unit IV	**Memory Management**	**08**
	1. The Process Address Space 2. Allocating Dynamic Memory 3. Managing Data Segment 4. Anonymous Memory Mappings 5. Advanced Memory Allocation 6. Debugging Memory Allocations 7. Stack-Based Allocations 8. Choosing a Memory Allocation Mechanism 9. Manipulating Memory 10. Locking Memory 11. Opportunistic Allocation (TextBook-3: Chapter8) 12. Swapping 13. Demand Paging (TextBook-1: Chapter Topics: 9.1, 9.2)	**01** **01** **01** **01** **01** **02** **01**
Unit V	**Signal Handling**	**10**
	1. Signal concepts 2. signal function 3. unreliable signal 4. interrupted system calls 5. reentrant functions 6. SIGCLD semantics 7. reliable-signal techno log i. kill and raise ii. alarm and pause iii. signal sets 8. sigprocmask i. sigpending ii. sigsetjmp and siglongjmp iii. sigsuspend	**02** **02** **01** **01** **04**

	vi. abort 9. system function revisited i. sleep (TextBook-2: Topics: 10.2-10.13,10.15-10.19)	

References:

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/ Paper Title	**Programming with DOT NET**
Course Code	**21SMCS122**
Semester	II
No. of Credits	04

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To understand the DOT NET framework
2.	To understand C# language features
3.	To understand Web development using ASP.NET

Expected Course Specific Learning Outcome

Sr. No.	Learning Outcome
1.	Student will able to develop application using DOT NET
2.	Student will able to develop application using C#
3.	Student will able to build web application using ASP.NET

Syllabus

Unit No	Title with Contents	No. of Lectures
	Part I : C#	
Unit I	**DOTNET Framework**	**10**
	1. Introduction to DOTNET 2. DOT NET class framework 3. Common Language Runtime : i.Overview ii.Elements of .NET application iii.Memory Management iv.Garbage Collection	**02** **02** **06**
Unit II	**Introduction to C#**	**12**
	1. Language features : i.Variables and Expressions ii. type conversion iii.Flow Control iv.Functions v.Delegates vi.Debugging and error handling v.exception handling (System Defined and User Defined) 2. Object Oriented Concepts i.Defining classes ii.class members iii.Interfaces, properties iv.Access modifiers v.Implementation of class vi.interface and properties vii.Overriding ix. Event Handling 3. Collections, Comparisons and Conversions i.Defining and using collections ii.Indexers, iterators iii.Type comparison iv.Value Comparison v.Overloading	**04** **04** **04**

Unit III	**Window Programming**	**10**
	1. Window Controls i.Common Controls ii.Container Controls (Group box and Tab controls) iii.Menus and Toolbars iv.Printing v.Dialogs 2. Deploying Window Application: i.Click Once deployment	**06** **04**
Unit IV	**Data Access**	**06**
	1. File System Data 2. XML 3. Databases and ADO.NET **4.** Data Binding	**02** **02** **02**
	Part II : ASP . NET	
Unit I	**Introduction to ASP.NET**	**08**
	1. Control Structures & Functions : 2. Forms, web pages, HTML forms, Web forms 3. Request & Response in Non-ASP.NET pages 4. Using ASP.NET Server Controls 5. Overview of Control structures 6. Functions : web controls as parameters	**01** **01** **01** **01** **02** **02**
Unit II	**Even Driven Programming and Post Back**	**04**
	1. HTML events 2. ASP.NET page events 3. ASP.NET Web control events 4. Event driven programming and post back	**02** **01** **01**
Unit III	**Reading from Databases**	**04**
	1. Data Source and Data binding controls 2. ADO.NET	**02** **02**

Unit IV	**ASP.NET Server Controls**	**06**
	1. ASP.NET Web Controls 2. HTML Server Controls 3. Web Controls	**02** **02** **02**

References:

1. Beginning Visual C#, Skinner, Kemper, Nagel, Wrox Publication
2. Professional C#, Nagel, Glynn, Skinner, Wrox Publication
3. Beginning ASP.NET 3.5, Jesse Liberty, Dan Hurwitz, and Dan Maharry, Wrox Publication
4. Programming ASP.NET 3.5, Jesse Liberty, Dan Maharry, Dan Hurwitz, O'Reilly Publication

Course/ Paper Title	**Software Project Management**
Course Code	**21SMCS123**
Semester	02
No. of Credits	04

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To covers skills that are required to ensure successful medium and large scale software projects.
2.	To examines Requirements Elicitation, Project Management, Verification &Validation and Management of Large Software Engineering Projects.
3.	To select and apply project management techniques for process modeling, planning, estimation, process metrics and risk management

Expected Course Specific Learning Outcomes

Sr. No.	Learning Outcome
1.	Student will able to collect requirements of project.
2.	Student will able to perform verification and validation of software projects
3.	Student will able to select particular technique for project management.
4.	Student will able to apply selected technique for project.

Syllabus

Unit No.	Title with Contents	No. of Lectures
Unit I	**Introduction to Project Management**	**05**
	1. What is a Project? 2. What is Project management? Project phases and project life cycle Organizational structure 3. Qualities of Project Manager 4. WBS	**01** **01** **01** **02**
Unit II	**Project Management Components**	**07**
	1. Project Integration Management-Project plan development and execution 2. Change control 3. CCB 4. Configuration management	**02** **02** **02** **01**
Unit III	**Scope Management**	**04**
	1. Strategic planning 2. Scope planning 3. Definition 4. Verification and control	**01** **01** **01** **01**
Unit IV	**Time management**	**02**
	1. Activity planning 2. Schedule development and control 3. GANTT Chart	**01** **01**

Unit V	**Cost Management**	**04**
	1. Cost estimation and Control 2. COCOMO model 3. BASIC COCOMO NUMERICALS	**01** **03**
Unit VI	**Quality Management**	**02**
	Quality planning and assurance	**02**
Unit VII	**Human Resource Management**	**02**
	1. Organizational planning **2.** Staff acquisition	**01** **01**
Unit VIII	**Communication Management**	**02**
	1. Information distribution **2.** Reporting	**01** **01**
Unit IX	**Risk Management**	**02**
	1. Risk identification **2.** Quantification and control	**01** **01**
Unit X	**Procurement Management**	**02**
	1. Solicitation management and control **2.** Contract administration	**01** **01**
Unit XI	**Software Metrics**	**07**
	1. The scope of software metrics 2. Size- oriented metrics 3. Function oriented 4. Software metrics data collection **5.** Analyzing software data	**01** **02** **01** **01** **02**
Unit XII	**Software Reliability**	**06**
	1. Measurement and prediction 2. Resource measurement **3.** Productivity, teams and tools	**02** **02** **02**
Unit XIII	**Planning a measurement program**	**05**
	1. What is metrics plan? 2. Developing goals, questions and metrics 3. Where and When: Mapping measures to activities 4. How: Measurement tools	**02** **02** **01**

	5. Who: Measurers , analyst, tools revision plans	
Unit XIV	**Quality Standards**	**04**
	1. CMM levels 2. KPA's **3.** PSP/TSP	**02** **01** **01**
Unit XV	**Introduction to DevOps**	**06**
	1. What is DevOps **2.** Why DevOps **3.** Principles of DevOps **4.** History of DevOps **5.** Architecture of DevOps **6.** Life Cycle of DevOps **7.** Overview of DevOps Tools **8.** How to achieve DevOps? **9.** DevOps and Agile	**02** **02** **02**

References :

1. Software Engineering, Roger Pressman, McGraw-Hill
2. Software Metrics for Project Management and process improvement, Robert B. Grady, Prentice hill

Course/ Paper Title	**Project**
Course Code	21SMCS124A
Semester	II
No. of Credits	02

Aims & Objectives of the Course

Sr. No.	**Objectives**
1.	To allow students to demonstrate the personal abilities and skills required to produce and present an extended piece of work
2.	To allow students to engage in personal inquiry, action and reflection on specific topics and issues.

3.	To allow students to focus on, and demonstrate an understanding of, the areas of interaction.
4.	To allow students to reflect on learning and share knowledge, views and opinions.

Expected Course Specific Learning Outcomes

Sr. No.	Learning Outcome
1.	Students will have abilities and skills skills required to produce and present an extended piece of work in corporate sectors.
2.	Students will know how to interact with team members while working on project.
3.	Students will able to share their knowledge and views.

Syllabus

Unit No.	Title with Contents	No. of Sessions
	Guidelines: • Students should work in a team of minimum 2 and maximum 3 students. • Students can choose a project topic without any restriction on technology or domain. • The student group will work independently throughout the project work including: problem identification, information searching, literature study, design and analysis, implementation, testing, and the final reporting. • Project guide must conduct project presentations (minimum 2) to monitor the progress of the project groups. • At the end of the project, the group should prepare a report which should conform to international academic standards. The report should follow the style in academic journals and books, with clear elements such as: abstract, background, aim, design and implementation, testing, conclusion and full references, Tables and figures should be numbered and referenced to in	15

	the report. • The final project presentation with demonstration (UE) will be evaluated by the project guide (appointed by the college) and one external examiner (appointed by the University).	

Course/ Paper Title	**Project Related Assignment**
Course Code	21SMCS125A
Semester	II
No. of Credits	02

Aims & Objectives of the Course

Sr. No.	**Objectives**
1.	The project assignments are a compulsory part of the project course and should be carried out by each project group.
2.	Project assignments are to be given by the guide for continuous internal evaluation.
3.	The project assignments are to be allotted to each group separately by the project guide on the basis of the implementation technology.

Expected Course Specific Learning Outcomes

Sr. No.	Learning Outcome
1.	Student will able to understand the flow of system development
2.	Student will able to form the content of documentation
3.	Student will able to understand documentation of testing of a project

Syllabus

Unit No.	Title with Contents	No. of Practical Sessions
UNIT I	Project Time management: plan (schedule table), Gantt chart, Roles and responsibilities, data collection, Implementation	15
	Simple assignments to evaluate choice of technology	
	Assignments on UI elements in chosen technology	
	Assignments on User interfaces in the project	
	Assignments on event handling in chosen technology	
	Assignments on Data handling in chosen technology	
	Online and offline connectivity	
	Report generation	
	Deployment considerations	
	Test cases	

Course/ Paper Title	**Human Computer Interaction**
Course Code	**21SMCS124B**
Semester	II
No. of Credits	02

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To design effective dialog for HCI.
2.	To design effective HCI for individuals and persons with disabilities.
3.	To assess the importance of user feedback.
4.	To develop meaningful user interface.

Expected Course Specific Learning Outcomes

Sr. No.	Learning Outcome
1.	Student will able to design effective dialog for HCI
2.	Student will understand importance of user feedback
3.	Student will able to develop meaningful user interfaces.

Syllabus

Unit No.	Title with Contents	No. of Lectures
Unit I	**FOUNDATIONS OF HCI**	**06**
	1. The Human: I/O channels i. Memory ii. Reasoning and problem solving 2. The computer: i. Devices ii.Memory iii. processing and networks; **3.** Interaction: i.Models ii. frameworks iii. Ergonomics iv. styles v. elements vi. Interactivity vii. Paradigms.	**01** **02** **03**
Unit II	**DESIGN & SOFTWARE PROCESS**	**07**
	1. Interactive Design basics i. process ii. scenarios iii. navigation viii. screen design ix. Iteration and prototyping. 2. HCI in software process i. software life cycle ii. usability engineering iv. Prototyping in practice v. Iv.design rationale. **3.** Design rules i. principles ii.standards iii. guidelines iv. rules **4.** Evaluation Techniques i.Universal Design	**02** **02** **02** **01**
Unit III	**MODELS AND THEORIES**	**05**
	1. Cognitive models i. Socio-Organizational issues and stake holder	 **01**

	requirements ii. Communication and collaboration models iii. Hypertext, Multimedia and WWW.	 **02** **02**
Unit IV	**MOBILE HCI**	**06**
	1. Mobile Ecosystem: i. Platforms ii. Application frameworks 2. Types of Mobile Applications: i.Widgets ii. Applications iii. Games iv.Mobile Information **3.** Architecture, Mobile 2.0, Mobile Design: i.Elements of Mobile Design ii. Tools.	**02** **02** **02**
Unit V	**WEB INTERFACE DESIGN**	**06**
	1. Designing Web Interfaces i. Drag & Drop ii. Direct Selection iii. Contextual Tools iv. Overlays, Inlays and Virtual Pages, v.Process Flow 2. Case Studies.	 **04** **02**

References :

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

Course/ Paper Title	**Human Computer Interaction Practical**
Course Code	**21SMCS125B**
Semester	II
No. of Credits	02

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To design effective dialog for HCI.
2.	To design effective HCI for individuals and persons with disabilities.
3.	To develop meaningful user interface.

Expected Course Specific Learning Outcomes

Sr. No.	Learning Outcome
1.	Student will able to design effective dialog for HCI
2.	Student will able to develop meaningful user interfaces.
3.	Student will able to design logo with effective user interface

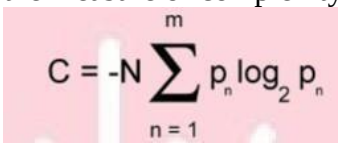
Syllabus

Unit No.	Title with Contents	No. of Practical Sessions
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 andreason.	15
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: Children (4-5 years of age): An application to teachmath. Perform analysis of children behavior e.g. their preferences, interests etc 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 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 Rural people: ATVM for train ticketing in ruralarea	

<table>
<tr><td></td><td>Perform analysis of rural people e.g. their problems, interests, needs, language etc
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.</td><td></td></tr>
<tr><td>3.</td><td>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.

<table>
<tr><td>Sr. No</td><td>Principles</td><td>Poor</td><td>Average</td><td>Good</td><td>Good Very</td><td>Excellent</td></tr>
<tr><td>1.</td><td>Aesthetically pleasing</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.</td><td>..</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td></td></tr>
<tr><td>4.</td><td>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 orcolumnar
Calculate Screen Complexity for existing Graphical User Interface(GUI).
Redesign the Screen by applying various guidelines to lower the complexity of selected Graphical User Interface (GUI) to achievesimplicity

Method for Measuring Complexity:
Draw a rectangle around each element on a screen, including captions, controls, headings, data, title, and soon.
Count the number of elements and horizontal alignment points (the number of columns in which a field, inscribed by a rectangle, starts).
Count the number of elements and vertical alignment points (the</td><td></td></tr>
</table>

	number of rows in which an element, inscribed by a rectangle, starts). 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.  $C = -N \sum_{n=1}^{m} p_n \log_2 p_n$ 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) Calculate overall complexity by adding the number bits required by horizontalalignment points and vertical alignment points.	
5.	Design/Redesign web user interface based on Gestalt theories and comment on the principle applied and justify. Also analyze 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 pixelssquare. Looks as good in black and white as in color. Icon Size Supply in all standardsizes. 16 × 16pixels. 16- and 256-color versions. - 32 × 32pixels 16- and 256-color versions. - 48 × 48pixels 16- and 256-colorversions. Use colors from the systempalette. Use an odd number of pixels along eachside. Provides center pixel around which to focus design. Minimum sizes for easyselection: -With stylus or pen: 15 pixels square. With mouse: 20 pixelssquare. With finger: 40 pixels square. - Provide as large a hot zone as possible. ChoosingImages Use existing icons whenavailable. Use images for nouns, notverbs. Use traditionalimages. 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 assuch. 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 forlegibility. 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, ifnecessary. Register new icons in the system'sregistry. Create and maintain a registry of all system icons. Provide a detailed and distinctive description of all new icons.	

Course/ Paper Title	**Soft Computing**
Course Code	**21SMCS124C**
Semester	II
No. of Credits	02

Aims & Objectives of the Course

Sr. No.	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.
4.	To apply the process of approximate reasoning using Neuro- Fuzzy Modeling.
5.	To provide the mathematical background to carry out optimization using genetic algorithms.

Expected Course Specific Learning Outcomes

Sr. No.	**Learning Outcome**
1.	Students will able to design experiments on real life problems using Neural Learning Algorithm
2.	Students will able to analyze experiments on real life problems using Neural Learning Algorithm
3.	Students will able to perform experiments on real life problems using Neural Learning Algorithm

Syllabus

Unit No.	**Title with Contents**	**No. of Lectures**
Unit I	**Introduction to Soft Computing**	**02**
	1. Neural Networks: i.Definition ii. Advantages iii. Applications iv. Scope. 2. Fuzzy logic: i.Definition ii. Applications. 3. Genetic Algorithms: i.Definition ii.Applications.	**01** **01**
Unit II	**Neural Network**	**15**
	1. Fundamental Concept: i.Artificial Neural Network ii. Biological Neural Network,	**01**

	2. Brain vs. Computer i. Comparison Between Biological Neuron and Artificial Neuron (Brain vs. Computer) ii. Artificial Neurons, 3. Neural Networks and Architectures: i. Neuron Abstraction ii. Neuron Single Functions iii. Mathematical Preliminaries 4. Neural Networks Defined, Architectures: i. Feed forward and Feedback ii. Salient Properties of Neural Networks 5. Geometry of Binary Threshold Neurons and Their Networks: i. Pattern Recognition and Data Classification ii. Convex Sets iii. Convex Hulls and Linear Separability iv. Space of Boolean Functions v. Binary Neurons are Pattern Dichotomizers vi. Non-linearly Separable Problems vii. Capacity of a Simple Threshold Logic viii. Neuron Revisiting ix. the XOR Problem x. Multilayer Networks xi. How Many Hidden Nodes are enough? 6. Learning and Memory: i. An Anecodatal Introduction ii. Long Term Memory iii. The Behavioral Approach to Learning iv. The Molecular Problem of Memory v. Learning Algorithm vi. Error Correction and Gradient vii. Descent Rules viii. Learning Objective for TLNs ix. Pattern Space and Weight Space x. Linear Seperabilty xi. Hebb Network xii. Perceptron Network xiii. α- Least Mean Square Learning.	**01** **02** **02** **04** **05**
Unit III	**Fuzzy Set Theory**	**09**
	1. Brief Review of Conventional Set Theory 2. Introduction to Fuzzy Sets 3. Properties of Fuzzy Sets	**01**

	4. Operations on Fuzzy Sets	**01**
	5. Crisp Relation	
	6. Fuzzy Relation	**01**
	7. Tolerance and equivalence relation	
	8. Fuzzy Tolerance and equivalence relation	**01**
	9. Fuzzy Max-Min and Max-Product Composition	
	Membership Functions	**01**
	10. Fuzzification, Defuzzification to crisp sets	
	11. λ-Cuts for fuzzy Relations	**01**
	12. Fuzzy (Ruled-Based) system	**01**
	13. Graphical technique of inference	**01**
	14. Membership value assignment-Intuition	**01**
	15. Inference.	
Unit IV	**Genetic Algorithms**	**04**
	1. What are Genetic Algorithms?	
	2. Why Genetic Algorithms?	**01**
	3. Traditional Optimization and Search Techniques	
	4. Simple GA	**01**
	5. Terminologies and Operators in GA	
	i. Encoding	**02**
	ii.Selection	
	iii.Crossover	
	iv.Mutation	
	v.Search	
	vi.Termination	
	vii.Constraintsin GA	

References:

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

Course/ Paper Title	**Soft Computing Practical**
Course Code	**21SMCS125C**
Semester	II
No. of Credits	02

Aims & Objectives of the Course

Sr. No.	**Objectives**
1.	To implement Fuzzy operations using any Technology
2.	To generate an ability to design, analyze and perform experiments on real life problems using various Neural Learning Algorithms.
3.	To Build simple Artificial Neural Network

Expected Course Specific Learning Outcomes

Sr. No.	**Learning Outcome**
1.	Students will able to implement Fuzzy operations.
2.	Students will able to analyze experiments on real life problems using Neural Learning Algorithm
3.	Students will able to perform experiments on real life problems using Neural Learning Algorithm.

Syllabus

Unit No.	Title with Contents	No. of Practical Sessions
UNIT I	Write a program to implement Fuzzy Operations Union Intersection Complement Algebraic sum Algebraic product Cartesian product	15
	Write a program to implement De Morgans law.	
	Write a program to implement Max-Min Composition and Max-Product Composition.	
	Write a program to implement lambda cut	
	Write a program to implement Activation Function.	
	Write a program to implement Perceptron Learning Rule	
	Write a program to implement Hebb's Rule	
	Write a program to implement Feed Forward Network	
	Write a program for building an Artificial Neural Network by implementing the Back propagation Algorithm and test the same using appropriate data sets.	
	Write a program for solving linearly separable problem using Perceptron Model.	
	Write a program to develop supervised learning algorithm	
	Write a program to study and analyze genetic life cycle	

Course/ Paper Title	**Practical on Advanced OS &Programming with DOT NET**
Course Code	**21SMCS126**
Semester	II
No. of Credits	04

Aims & Objectives of the Course

Sr. No.	Objectives
1.	To get familiar with the Shell commands on LINUX in AOS.
2.	To get the knowledge of file handling using LINUX commands.
3.	To familiar with the functions and Framework of DOT NET Technology.
4.	To build a simple application using DOT NET Framework

Expected Course Specific Learning Outcomes

Sr. No.	Learning Outcome
1.	Student will be familiar with the Shell commands on LINUX using AOS.
2.	Student will get the knowledge of file handling using LINUX

	commands.
3.	Student can build a simple application using DOT NET Framework

Syllabus

Unit No.	Title with Contents	No. of Sessions
	LIST OF AOS PROGRAMS	15
UNIT I	To create 'n' children. When the children will terminate, display total cumulative time children spent in user and kernel mode.	
	To generate parent process to write unnamed pipe and will read from it.	
	To create a file with hole in it.	
	Takes multiple files as Command Line Arguments and print their inode number.	
	To handle the two-way communication between parent and child using pipe.	
	Print the type of file where file name accepted through Command Line.	
	To demonstrate the use of at exit () function.	
	Open a file goes to sleep for 15 seconds before terminating.	
	To print the size of thefile.	

	Read the current directory and display the name of the files, no of files in currentdirectory.	
	Write a C program to implement the following unix/linux command (use fork, pipe and exec system call) ls –l | wc –l	
	Write a C program to display all the files from current directory which are created in particular month	
	Write a C program to display all the files from current directory whose size is greater that n Bytes Where n is accept fromuser.	
	Write a C program to implement the following unix/linux command ls –l > output.txt	
	Write a C program which display the information of a given file similar to given by the unix /linux command ls –l <filename>	
	Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell$". Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the followingcommand. count c <filename> - print number of characters in file count w <filename> - print number of words in file count l <filename> - print number of lines in file	
	Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell$". Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command. i. list f<dirname> - print name of all files in directory ii. list n <dirname> - print number of all entries iii. list i<dirname> - print name and inode of all files	
	Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell$". Any normal shell command is executed from your shell by	

	starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command. i. typeline +10 <filename> - print first 10 lines of file ii. typeline -20 <filename> - print last 20 lines of file iii. typeline a <filename> - print all lines of file	
	Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell$".Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command. i. search f <pattern><filename> - search first occurrence of pattern in filename ii. search c <pattern><filename> - count no. of occurrences of pattern in filename iii. search a <pattern><filename> - search all occurrences of pattern in filename	
	Write a C program which receives file names as command line arguments and display those filenames in ascending order according to their sizes. i) (e.g $ a.out a.txt b.txt c.txt, …)	
	Write a C program which creates a child process which catches a signal sighup, sigint and sigquit. The Parent process send a sighup or sigint signal after every 3 seconds, at the end of 30 second parent send sigquit signal to child and child terminates my displaying message "My DADDY has Killed me!!!".	
	Write a C program to implement the following unix/linux command (use fork, pipe and exec system call). Your program should block the signal Ctrl-C and Ctrl-\ signal during the execution. ls –l | wc–l	
	Write a C Program that demonstrates redirection of standard output to a file.	
	Write a program that illustrates how to execute two commands concurrently with a pipe.	

	Write a C program that illustrates suspending and resuming processes using signals.	
	Write a C program that illustrates inter process communication using shared memory.	
	Programming with DOT NET	15
UNIT I	Write a program to work with String Builder • Create a string Assign it with large string value consisting of no of words • Access the string character by character and print • Access the string word by word and print • Find a pattern in the string such as "AB" and replace it with some other string	
	Write a program to implement Custom Exception. Create Invalid Student Name Exception class in a school application, which does not allow any special character or numeric value in a name of any of the students. Use Reg ex("^[a-z A-Z]+$") to check Student Name	
	Write a form based program offering binary calculator having following functionality • Add, multiply, subtract, divide • Left shift, right shift	
	Write a program which implements following classes Write a class Earth (Producer), which exposes static event Earth Quake Implement classes hospital, NGO who respond to Earth Quake event Execution: • On Click of a button on Form, Earth Quake event should be triggered. • Message should be shown that NGO and Hospital have responded to it	
	To implement reflection do following a. Implement a class library as follows. • Car class - 2 methods, 2 member variables Write an application (Console based) performing following tasks using reflection • Load Class library using reflection • Iterate class – types, display type detail	
	Create base class Customer and subclasses Silver Customer and Gold Customer • Define discount() method in Customer class which returns 20% discount Overload discount method in the subclasses and return different discount value Define base class variable as "Customer cust" Assign different objects of Customer, Silver Customer and Gold	

	Customer to variable cust one after other and invoke discount method each time. What is the discount % returned each time?	
	Design a form which offers User Registration Form On Click of OK, registered user data should get saved in XML	
	Write a program to create a magic square using Windows Forms? Accept square dimension from user.	
	Implement a Simple Editor which has following features • Menu : File, New, Save, Print Preview, Print • Toolbar: Formatting for Bold, Italic, Underline	