Modern Education Society's

Nowrosjee Wadia College

(AUTONOMOUS)

NAAC Accredited A+ with CGPA 3.51

Affiliated to the
Savitribai Phule Pune University
(Formerly University of Pune)

Two Year M.Sc. Degree Program in Computer Science

M.Sc. (Computer Science)

(Faculty of Science & Technology)

National Education Policy (NEP) Syllabus
To be implemented from Academic Year 2023-2024

Preamble of the Syllabus:

MSc Computer Science is a two-year postgraduate programme that aims to develop advanced programming skills and strategies for solving various logical challenges using various current trends in IT industry. The course also provides a necessary foundation for students aspiring to obtain deeper knowledge in the domain of Computer Science, through Research and Inventions. 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 proposed choice-based credit curriculum and grading system designed as per NEP, offers a wide range of Electives as per Industry requirements / Research requirements, to choose from, during the 4-semester Course schedule. The Major core subjects and the Electives are designed keeping in view the expanse and applications of Computer Science in various real-life domains. The post graduate program is aimed to cater to this ever-increasing demand and groom the students to excel in their future career, in the IT industry.

Introduction:

The syllabus for the Masters in Computer Science course syllabus is revised as per NEP guidelines 2020, to meet the needs of dynamically changing technological scenarios in Computer science. The changing scenario of higher education in India and abroad is taken into consideration while formulation this syllabus and more oriented towards current need of modern Research and Industrial sectors.

The present syllabus is as per National Education Policy 2020, which amalgamates the fundamental Core subjects at one end and the latest technologies in Computer science at the other. The Major core subjects and the Electives are provided with the main focus on improving and strengthening a student’s knowledge on Core fundamentals of Computer science domain, as well as improve their expertise in the ever changing latest technologies in the Computer Science domain. Extensive practical courses are designed to supplement the theory courses with hands on experimentation in the Laboratory. Empowerment of students to face Research and Industrial needs is at the centre of this syllabus.

Through Electives, students are provided with a wide range of subjects in this domain, to be chosen as per their area of interest and the current Industry needs. This will help a student develop the depth in specialization and make them ready to face the upcoming technological advances in the world without any further training. M.Sc. syllabus has been prepared keeping in vision the undergraduate curriculum.. Field Projects / Research project modules are incorporated into the current syllabus, to provide a buffer zone for the budding Software Engineers / Computer Science domain based Researchers eager to enter the Computer Science sector.

Course Objectives:

1. To enhance abilities of students in problem solving using computers.
2. To upgrade the necessary skill set and analytical abilities for developing computer-based solutions for real life problems.
3. To provide the necessary foundation base for research and development in computer science.
4. To enhance the professional skills of the students.
5. To enable students build a successful career in Computer Science and t produce entrepreneurs who develop software products.

Eligibility for the course M.Sc. Computer Science.

Any candidate completed B.Sc. in Computer Science from any recognized university.

Examination pattern

70 Marks for end semester examination and 30 marks for continuous evaluation pattern

35 Marks for External Practical examination and 15 Marks for Internal practical examination

Passing marks

Passing marks will be 40% in each paper, of continuous evaluation, and End Semester Exam separately.

Procedure for continuous evaluation

Written test	20 marks
Assignment	5 Marks
Seminar/ attendance	5 Marks
	30 Marks

Nature of question paper for End semester examination

Que No.	Type	Max marks
1	2 marks x 7 questions (Any 6)	12
2	4 marks x 4 questions (Any 3)	12
3	5 marks x 4 questions (Any 3)	15
4	5 marks x 4 questions (Any 3) Short notes type	15
5	8 marks x 3 questions (Any 2)	16
	Total	70 marks

Revaluation

There shall be revaluation of answer scripts of end semester examination (out of 70 marks) of theory papers only, but not of internal or continuous evaluation papers as per Ordnance no. 134 A and B

Grading system

Marks	Grade	Grade Point
80-100	O: Outstanding	10
70-79	A+: Excellent	9
60-69	A: Very Good	8
55-59	B+: Good	7
50-54	B: Above Average	6
45-49	C: Average	5
40-44	P: Pass	4
0-39	F: Fail	0
-	Ab: Absent	0

Course structure semester I

Year	Semester	Course Type	Choice	Course code and Course Name	Credits
1	1	Major	Mandatory	Principles of Programming Languages	**4**
				Database Technologies	**4**
				Laboratory Course on Principles of Programming Languages & Database Technologies	**2**
				Laboratory course on Advanced Operating Systems	**2**
			Electives	Mobile application development Laboratory course on Mobile application development **OR** Design and analysis of algorithms (DAA) Laboratory course on DAA (SciLab)	**2** **2** **2** **2**
		RM		Research methodology	**4**
		OJT / FP		NA	**-**
		RP		NA	**-**
Total credits					**20**

Course structure semester II

Year	Semester	Course Type	Choice	Course code and Course Name	Credits
1	2	Major	Mandatory	Web Frameworks	**4**
				Artificial Intelligence & Machine Learning	**4**
				Laboratory Course on Web frameworks	**2**
				Laboratory Course on Artificial Intelligence & Machine learning	**2**
			Electives	Dot Net Laboratory course on Dot NET **OR** Software Project management Laboratory course on SPM (Project management tool)	**2** **2** **2** **2**
		RM		NA	**-**
		OJT / FP		On Job training / field project	**4**
		RP		NA	**-**
Total credits					**20**

Course Code: PCSMJ111
Title : Principles of Programming Languages

Teaching Scheme	No. of Credits	Examination Scheme
60 Hours	4	CE : 30 marks SEE : 70 marks

Prerequisites

Student should have basic knowledge of:

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

Course Objectives:

1. To prepare students to think about programming languages analytically.
2. To separate syntax from semantics.
3. Compare programming language designs.
4. Understand their strengths and weaknesses.
5. Learn small programs in different programming languages.

Course Contents

Unit 1	**Introduction**	**3**

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

Unit 2	**Names, Scopes and Bindings**	**6**

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

Unit 3	**Control Flow**	**6**

3.1 Expression Evaluation , Precedence and Associativity, Assignments, Initialization,Ordering Within Expressions, Short-Circuit Evaluation 3.2 Structured and Unstructured Flow, Structured Alternatives to got 3.3 Sequencing 3.4 Selection - Short-Circuited Conditions, Case/Switch Statements Iteration 3.5 Iteration - Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion 3.6 Recursion - Iteration and Recursion, Applicative- and Normal-Order Evaluation		
Unit 4	**Data Types**	**13**
4.1 Introduction 4.2 Primitive Data Types 4.3 Numeric Types : Integer, Floating point, Complex , Decimal, Boolean Types,Character Types 4.4 Character String Types 4.5 Design Issues, Strings and Their Operations, String Length Operations, Evaluation, Implementation of Character String Types 4.6 User defined Ordinal types Enumeration types, Designs Evaluation Subrange types,Ada's design Evaluation Implementation of user defined ordinal types 4.7 Array types 4.8 Design issues, Arrays and indices, Subscript bindings and array categories, Heterogeneous arrays, Array initialization, Array operations, Rectangular and Jaggedarrays, Slices, Evaluation, Implementation of Array Types 4.9 Associative Arrays 4.10 Structure and operations, Implementing associative arrays 4.11 Record types 4.12 Definitions of records, References to record fields, Operations on records, Evaluation, Implementation of Record types 4.13 Union Types 4.14 Design issues,Discriminated versus Free unions,Evaluation,Implementation of Union types 4.15 Pointer and Reference Types 4.16 Design issues, Pointer operations, Pointer problems, Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation 4.17 Implementation of pointer and reference types - Representation of pointers and references Solution to dangling pointer problem Heap management		
Unit 5	**Subprograms and Implementing Subprograms**	**6**
5.1 Introduction 5.2 Fundamentals of Subprograms 5.3 Design Issues for subprograms 5.4 Local Referencing Environments 5.5 Parameter-Passing Methods 5.6 Parameters that are Subprograms 5.7 Overloaded Subprograms 5.8 Generic Subroutines, Generic Functions in C++, Generic Methods in Java		

5.9 Design Issues for Functions 5.10 User-Defined Overloaded Operators 5.11 Coroutines 5.12 Implementing Subprograms 5.13 The General Semantics of Calls and Returns 5.14 Implementing "Simple" Subprograms 5.15 Implementing Subprograms with Stack- Dynamic Local Variables 5.16 Nested Subprograms 5.17 Blocks 5.18 Implementing Dynamic Scoping		
Unit 6	**Data Abstraction and Object Orientation**	**11**
6.1 Object-Oriented Programming 6.2 Encapsulation and Inheritance- Modules, Classes, Nesting Inner Classes), Type Extensions, Extending without Inheritance 6.3 Initialization and Finalization- Choosing a Constructor, References and Values,Execution Order, Garbage Collection 6.4 Dynamic Method Binding 6.5 Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures 6.6 Multiple Inheritance 6.7 Semantic Ambiguities, Replicated Inheritance Shared Inheritance, Mix-In Inheritance 6.8 Concurrency**-** Introduction : Multiprocessor Architecture Categories of concurrency, Motivations for studying concurrency. Introduction to Subprogram-level, concurrency Fundamental concepts, Language Design for concurrency, Design Issues		
Unit 7	**Functional Programming in Scala**	**15**
7.1 Strings 7.2 Numbers 7.3 Control Structures 7.4 Classes and Properties 7.5 Methods 7.6 Objects 7.7 Functional Programming 7.8 List,Array, Map, Set		
Learning Outcomes 1. Students will be able to learn new languages more quickly. 2. Students will be able to understand basic language implementation techniques. 3. Students will be able to write simple Scala programs.		

Learning Resources

1. Programming Language Pragmatics, Michel L. Scott, Kaufman Publishers,An Imprint of Elsevier, USA, 3rd Edition
2. Concepts of Programming languages, Robert W. Sebesta, Pearson Education,Eighth edition
3. Scala Cookbook, Alvin Alexander ,O'REILLY publication

Course Code: PCSMJ112
Title : Database Technologies

Teaching Scheme	No. of Credits	Examination Scheme
60 lectures	4	CE : 30 marks ESE: 70 marks

Prerequisites :

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

Course Objectives :

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

Course Contents

Chapter 1	**Introduction to NOSQL (Core concepts)**	**18**
1.1 Why NoSQL 1.2 Aggregate Data Models 1.3 Data modeling details 1.4 Distribution Models 1.5 Consistency 1.6 Version stamps Map-Reduce		
Chapter 2	**Implementation with NOSQL databases**	**14**
2.1 Key-Value Databases (Riak) 2.2 Document Databases (Mongodb) 2.3 Column-Family stores (Cassandra) 2.4 Graph databases (Neo4j)		
Chapter 3	**Schema Migrations**	**6**
3.1 Schema Changes 3.2 Schema Changes in RDBMS 3.3 Schema Changes in a NoSQL Data Store		
Chapter 4	**Polygot Persistence (Multi model types)**	**8**
4.1 Disparate Data Storage Needs 4.2 Polyglot Data Store Usage 4.3 Service Usage over Direct Data Store Usage 4.4 Expanding for Better Functionality 4.5 Choosing the Right Technology 4.6 Enterprise Concerns with Polyglot Persistence 4.7 Deployment Complexity		
Chapter 5	**Beyond NoSQL**	**8**
5.1 File Systems 5.2 Event Sourcing 5.3 Memory Image 5.4 Version Control 5.5 XML Databases 5.6 Object Databases		
Chapter 6	**Choosing your database**	**6**
6.1 Programmer Productivity 6.2 Data-Access Performance		

6.3 Sticking with the Default
6.4 Hedging Your Bets

Learning Outcomes:

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

Learning Resources

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

Course Code: PCSMJ113

Title : Laboratory Course on Principles of Programming Languages and Database Technologies

Teaching Scheme	No. of Credits	Examination Scheme
4 hours / week per batch	2	CE : 15 marks ESE: 35 marks

Prerequisites:

1. Basic Knowledge of DBMS and SQL.
2. Knowledge on Java Programming Language.

Course Objectives :

1. To write concise code and to learn multi-paradigm aspects of Scala programming Language.
2. To perform CRUD operations, Data modelling and aggregation in Mongodb and Neo4j.

Assignments :

MongoDB Assignments

1. Create a 'Movie' database with the collections 'film' and 'actor'. Add required fields and insert 10 documents in each collection. Perform CRUD operations.
2. Create a 'Company' database with the collections 'employee' and 'transactions' . Add required fields and insert 10 documents in each collection. Perform CRUD operations.
3. Perform queries based on 'Company' database having collections 'Employee' and 'Transactions'.
   [Read MongoDB Aggregate framework and Map Reduce before executing the assignment.]
   Note: It is expected that students should fill in the data relevant to the queries given in the assignment. The result set should not be empty.

Neo4j Assignments

1. Create the following databases as graph models.
   A) Library
   B) Song
   C) Employee
   D) Movie
   Visualize the models after creation, Return properties of nodes, Return the nodes labels, Return the relationships with its properties.
   Note: You may assume and add more labels , relationships, properties to the graphs
2. Perform Simple queries based on the graph models created in assignment 1.
3. Perform Complex queries based on the graph models created in assignment 1.

Scala Assignment 1: 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 the factorial of a number.
3. Write a program to read five random numbers and check that random numbers are perfect or not.
4. Write a program to find the 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 the user and convert it to binary and octal using user defined functions.

Scala Assignment 2: Strings

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 the user and count the number of occurrences of that character.
3. Write a program to read two strings. Remove the occurrence of the second string in the first string.
4. Create an array of strings and read a string from the user. Display all the elements of array containing given string.

Learning Outcomes :

On completion of the course, student will be able to:

1. Get introduced to Scala and implement features of Scala.
2. Understand the working of Mongodb and Neo4j features.

Course Code: PCSMJ114
Title: Laboratory Course on Advanced Operating System

Teaching Scheme : 4 hrs/week per batch	No. of Credits 2	Examination scheme: IA: 15 Marks UE: 35 Marks

Course Prerequisites:

1. Basic C programming, algorithms and data structure concepts
2. Basic Computer Architecture concepts.
3. Basic Operating System Concepts
   (Process creation, scheduling and synchronization)
   Memory management(Paging, logical, physical, virtual address spaces, allocation methods)

Course Objectives:

1. To provide a deep understanding of the Linux Operating system internals, through self explanatory theory and hands-on practicals
2. This course provides an understanding of the functions and functional modules of Operating Systems.
3. To provide an introduction to the advanced O.S concepts like virtualization / Zoning and Containerization.
4. To provide the core foundations to further study Cloud computing.

Assignment 1	**File and Directory I/O**	**No. of slots : 02**

- Programs based on File and Directory I/O system calls related to various algorithms on file subsystems in Linux.
- Pre-reading
  1. Concept of file descriptor
  2. File status flags
  3. Stat structure for a file
  4. File types and File access permissions
  5. Dirent structure for directory
- Algorithms
  1. Open
  2. read
  3. write
  4. pipe
- Assignments
  1. Write a program to create a file with a hole in it.
  2. Write a program to open a file and go to sleep for 15 seconds before terminating.
  3. Write a program to read the current directory and display the name of the files, size of the file, type of file and no of files in the current directory.

Assignment 2	**Buffer block and free list allocation**	**No. of slots : 02**

- Programs based on Buffer block allocation and free list allocation related to various buffer allocation algorithms.
- Pre-reading
  1.Buffer allocation scenarios
  2. Data Structure (Linked list)
- Algorithms
  1. bread, breada, bread
  2. bwrite, getblk, brelse
  3. namei, iget, iput
  4. ialloc, ifree
- Assignments

Write a C program that illustrates banking transactions using the 4th and 5th buffer allocation scenario. Consider three processes- EMI, withdraw and deposit.
EMI and withdrawal processes will go into sleep mode due to insufficient balance. These two processes should be included in the race condition after completion of the deposit process. Display interprocess communication between all these processes by implementing free list and buffer block list. (Use appropriate system calls too).

Assignment 3	**Process Environment & Control**	**No. of slots : 02**

- Programs based on the processing environment of UNIX system calls and related various algorithms.
- Pre-reading
  1. Processing environment of unix , Process identifiers
  2. Concept of sharing of variables between parent and child process
  3. Syntax, use and implementation of functions: atexit, getenv, putenv, setenv, unsetenv, getrlimit, fork.
- Algorithms
  1. fork
  2. issig, psig
  3. exit, atexit
  4. wait, exec, pipe
- Assignments
  1. Write a program to create 'n' children. When the children will terminate, display total cumulative time children spent in user and kernel mode.
  2. Write a program to demonstrate the use of atexit() function.
  3. Write a program to handle the two-way communication between parent and child using pipe.

Assignment 4	**Memory Management**	**No. of slots : 02**

- Programs based on Memory management system calls and related various algorithms in Linux.
- Pre-reading
  1. Memory Allocation Mechanism
  2. Manipulating Memory
- Algorithms
  1. malloc, alloc, free
  2. shmat, semop
  3. fork, vfork, etc.
- Assignments

Write a C program that illustrates inter process communication using shared memory.

Assignment 5	**Signal Handling**	**No. of slots : 02**

- Programs based on Signal Handling in Linux based on various system calls and related algorithms.
- Pre-reading
  1. Concept of process accounting
  2. User identification by process
  3. Signal concepts
  4. Syntax, use and implementation of below function:
     system, exec functions, getlogin, kill, alarm, raise, pause, sigprogmask, sigsuspend, etc
- Algorithms
  1. issig, psig
  2. exit
  3. brk, exec
- Assignments
  1. 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!!!".
  2. Write a C program that illustrates suspending and resuming processes using signals.
  3. 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

Assignment 6	**Virtualization**	**No. of slots : 02**

- Programs based on Virtualization on Linux.
- Pre-reading
  1. Concept of Virtualization.
  2. Virtualization types : Server Virtualization, Client and Desktop Virtualization, Services and applications Virtualization.
  3. Hypervisor and its types.
- Assignments
  Assignments Based to creation of application virtualization and its working on client and server.

Assignment 7	**Containerization**	**No. of slots : 02**

- Programs based on Containerization on Linux using Dockers.
- Pre-reading
  1. Concept of Containerization and Dockers containers.
  2. Docker daemon, Docker client, Docker Desktop, Docker registries, Docker objects, etc.
  3. Deployment of Dockers.
- Assignments
  Assignments Based to creation of Docker containers on Linux and its deployment.

Assignment 8	**Zoning**	**No. of slots : 02**

- Programs and Activity based on Zoning.
- Pre-reading
  1. Concept of Containerization and Dockers containers.
  2. Concept of Zoning.
  3. Concept of SAN and its types.
- Assignments
  Assignments Based to creation of Zones and setting up Zone on your system.

Learning Outcomes

After completion of this course students will be able to -

1. Students will be able to implement various algorithms for various units of Unix OS.
2. Students will be able to apply various system calls to do the operations on OS.
3. Students will be able to apply different algorithms to handle the signals in process communications.
4. Students will be able to create VM, Docker containers and Zone on Linux OS.
5. Students will understand the necessary concepts which will help to migrate the Applications / services on cloud.

Course Code : PCSMJ115A
Title : Mobile Application Development

Teaching Scheme	No. of Credits	Examination Scheme
30	2	IE : 15 marks SEE: 35 marks

Prerequisites

1. Basic knowledge about programming
2. Learning of Java or Kotlin
3. Concepts of OOPs
4. Knowledge about any database management system
5. Understanding of XML

Course Objectives

1. To expose students to various aspects of mobile and ad-hoc networks.
2. To create robust mobile applications and learn how to integrate them with other services.
3. To Create intuitive,reliable mobile apps using the android services and components.
4. To create a seamless user interface that works with different mobile screens .
5. Understand Mobile security.

Course Contents

Chapter 1	Android Fundamentals	**6**

1.1 Mobile Operating system – IOS, Introduction to Android - Overview and evolution of Android , Features of Android, Android architecture
1.2 Components of an Android Application, Manifest file
1.3 Android Activity
1.4 Service Lifecycle

Chapter 2	Android UI Design	**6**

2.1 Basic UI Designing (Form widgets ,Text Fields , Layouts ,[dip, dp, sip, sp] versus px)
2.2 Intent(Implicit & Explicit)
2.3 All components (e.g Button , Slider,Image view, Toast)
2.4 Event Handling
2.5 Adapters and Widgets
2.6 Menu

Chapter 3	Android Thread and Notification	**6**

3.1 Threads running on UI thread (runOnUiThreadBuffer)
3.2 Worker thread
3.3 Handlers & Runnable
3.4 AsyncTask (in detail)
3.5 Broadcast Receivers
3.6 Services and notifications
3.7 Alarms

Chapter 4	Advanced Android Programming	**6**

4.1 Content Providers – SQLite Programming
4.2 JSON Parsing
4.3 Accessing Phone Service(Call, SMS, MMS)

4.4 Location based services

Chapter 5	PhoneGap Programming and IOS Basics	**6**

5.1 Why Use PhoneGap?
5.2 How PhoneGap Works
5.3 Designing for the Container
5.4 Writing PhoneGap Applications
5.5 Building PhoneGap Applications
5.6 PhoneGap Limitations
5.7 Introduction - What is IOS ,IOS Architecture
5.8 Frameworks , Application Life cycle , Features

Learning Outcomes

1. Students will understand how to develop the wireless mobile application in android using Java/Kotlin.
2. Students will get the practical knowledge of what it takes to create a mobile app and this will help them have a strong foundation of application development.
3. Students will be ready with the technology which is used widely in Mobile Industry as a part of Android developer.
4. Design application using open source technology.

Learning Resources:

1. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides), 2013 byBill Phillips and Brian Hardy
2. Professional Android 4 Application Development,2012 by Reto Meier
3. Android Application Development in 24 Hours, Sams Teach Yourself 2015 by Carmen Delessio and Lauren Darcey 4th edition.
4. Beginning Android Application Development Wei-Meng Lee Wiley Wrox publication Second edition
5. Android cookbook Ian F Darwin O'Reilly publication Second Edition
6. PhoneGap: Beginner's Guide Giorgio Natili, Purusothaman Ramanujam PACKT Publication Edition4.

Course Code : PCSMJ115B
Title : Laboratory Course on Mobile Application Development

Teaching Scheme 4 hrs/week per batch	No. of Credits 2	Examination Scheme IE : 15 marks ESE: 35 marks

Prerequisites : Mobile Technology

Course Objectives

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

List Of Assignment For Mobile Technology

Assignment 1 : Setting up of Android Studio and Emulator Basic Application programs

Assignment 2 : Use of Activity ,Intents,views,Fragments and Screen design using UI Components.

Assignment 3 : Programs based on Spinner ,Toast, Alert Box , Dialogs Box ,List and Adapter

Assignment 4 : Programs How to connect with SQLite Database and use of Multimedia Files in Android Application.

Assignment 5 : Programs Working with Google maps and Notification.

Learning Outcomes

After completion of this course students will be able to -

1. Install and configure Android application development tools.
2. Design and develop user Interfaces for the Android platform.
3. Apply Java programming concepts to Android application development.
4. Students will be able to Design android apps using location based services and other multimedia utility.

Course Code: PCSMJ116A

Title: Design and Analysis of Algorithms

Teaching Scheme	No. of Credits	Examination scheme:
30 Lectures	02	IA: 15 Marks UE: 35 Marks

Prerequisites:

1.Basic algorithms and data structure concepts

2.Basic Graph terminologies.

3.Basic Tree terminologies.

Course Objectives:

1.To design the algorithms
2.To select the appropriate algorithm by doing necessary analysis of algorithms
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 problems and solutions.
7.Learn a variety of useful algorithms.
8.Understand classification of problems.
9.To provide foundation in algorithm design and analysis.
10. To develop the ability to understand and design algorithms in the context of space and time complexity.

Course Contents

Unit 1	**Basics of Algorithms**	**06**

1.1 Algorithm definition and characteristics
1.2 Space complexity, Time complexity
1.3 worst case-best case-average case complexity asymptotic notation
1.4 Sorting algorithms basics : (insertion sort, heap sort, bubble sort)
1.5 Sorting algorithms (counting sort, concept of bucket and radix sort)
1.6 Searching algorithms: Linear, Binary

Unit 2	**Divide and conquer strategy**	**02**

2.1 Merge sort, Quick sort with analysis.

2.2 Comparison between Traditional Method of Matrix Multiplication vs. Strassen's Matrix Multiplication

Unit 3	**Greedy Method**	**04**

3.1 Knapsack problem. 3.2 Job sequencing with deadlines, 3.3 Minimum-cost spanning trees : Kruskal and Prim's algorithm 3.4 Optimal storage on tapes 3.5 Optimal merge patterns 3.6 Huffman coding 3.7 Shortest Path :Dijkstra's Algorithm		
Unit 4	**Decrease and Conquer and Backtracking**	**08**
4.1 Definition of Graph Representation of Graph 4.2 Terminologies : Topological sorting, Connected components, spanning trees, Articulation Point, Bridge edge, DFS, BFS 4.3 Fixed Tuple vs. Variable Tuple Formulation 4.4 4- Queens problem 4.5 Graph coloring problem 4.6 Hamiltonian cycle 4.7 Sum of subsets		
Unit 5	**Branch and Bound and Problem Classification**	**10**
5.1 Introduction : LCBB Search 5.2 Terminologies : FIFO BB Search, LIFO Search, Bounding Function, Ranking Function, 5.3 Traveling Salesman problem Using Variable tuple Formulation using LCBB 5.4 0/1 knapsack problem using LCBB 5.5 Nondeterministic algorithm 5.6 The class of P, NP, NP-hard and NP -Complete problems 5.7 Cook's theorem		
Learning Outcomes 1. Understanding Algorithmic complexity and analyzing the same. 2. Developing an understanding of various techniques and methods to design algorithms. 3. Skill to make the algorithm and solve real-world problems. 4. Analysis of traditional algorithms and apply to various problems.		
Learning Resources 1. Computer algorithms Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran Galgotia Publication 2. T. Cormen, C. Leiserson, & R. Rivest Algorithms MIT Press 3. 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 : PCSMJ116B

Title : Laboratory Course on Design and Analysis of Algorithms(SciLab)

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs/week per batch	2	IE : 15 marks ESE: 35 marks

Prerequisites : Design and Analysis of Algorithms

Course Objectives

1. To learn installation and configuration of Scilab console and tools.
2. To learn to implement various algorithms.
3. To learn analysis of algorithms in terms of complexity.

List Of Assignment For Mobile Technology

Assignment 1: Sorting and Searching algorithms analysis.

Programs based on Bubble sort, Insertion Sort, Linear and Binary Search

Assignment 2 : Divide and Conquer

Programs based on Merge sort, Quick Sort, Strassen's Matrix Multiplication

Assignment 3 : Greedy Method

Programs based on Job sequencing, Kruskal and Prim's algorithm.

Assignment 4 : Decrease and Conquer and Backtracking

Programs based on DFS, BFS, Graph coloring problem algorithms.

Assignment 4 : Branch and Bound and Problem Classification

Programs based on Traveling Salesman problem Using Variable tuple Formulation using LCBB, 0/1 knapsack problem using LCBB

Learning Outcomes

After completion of this course students will be able to -

1. Install and configure Scilab console and tools.
2. Design and develop various algorithms in Scilab.
3. Students will be able to apply different algorithms to solve real world problems and analyze them in terms of complexity.

Course Code: PCSRM117
Title : Research Methodology

Teaching Scheme	No. of Credits	Examination Scheme
60 lectures	4	CE : 30 marks ESE: 70 marks

Prerequisites

1. The actuality of the theme of the research
2. The choice of adequate research instruments and taxonomy to the chosen object field
3. Availability of a research capacity of the author.

Learning Objectives

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

Course Contents

Unit 1	**Introduction To Research Methodology**	**10**

1.1 Meaning of Research
1.2 Objectives of Research
1.3 Motivation in Research
1.4 Types of Research
1.5 Research Approaches
1.6 Significance of Research
1.7 Researcher and Characteristics of Researcher
1.8 Research Ethics and Integrity
1.9 Plagiarism and types of plagiarism
1.10 Introduction to Plagiarism check tools
1.11 Research Methods versus Methodology
1.12 Research and Scientific Method
1.13 Importance of Knowing How Research is Done
1.14 Criteria of Good Research

Unit 2	**Literature Review and Formulation of Research Problems**	**6**

2.1 Research Process
2.2 Reviewing the literature: purpose of a literature review

2.3 Literature resources 2.4 The Internet and a literature review 2.5 The Internet and research strategies and methods 2.6 Conducting and Evaluating literature reviews 2.7 Formulation of research problem 2.7.1 What is a Research Problem? 2.7.2 Selecting the Problem 2.7.3 Necessity of Defining the Problem 2.7.4 Technique Involved in Defining a Problem		
Unit 3	**Research Design**	**8**
3.1 Meaning of Research Design 3.2 Need for Research Design 3.3 Features of a Good Design 3.4 Important Concepts Relating to Research Design 3.5 Different Research Designs/Methods 3.5.1 Pure and Applied Research 3.5.2 Exploratory or Formulative Research 3.5.3 Descriptive Research 3.5.4 Diagnostic Research 3.5.5 Evaluation Studies 3.5.6 Action Research 3.5.7 Experimental Research 3.5.8 Analytical Study or Statistical Method 3.5.9 Historical Research 3.5.10 Surveys 3.5.11 Case Study 3.5.12 Field Studies		
Unit 4	**Hypothesis and Sampling**	**10**
4.1What is Hypothesis? 4.2 Nature & Characteristics of Hypothesis 4.3 Significance of Hypothesis 4.4 Types of Hypothesis 4.5 Sources of Hypothesis 4.6 Characteristics of Good Hypothesis 4.7 What is sampling? 4.8 Aims of Sampling 4.9 Characteristics of Good Sample 4.10 Basis of Sampling 4.11 Merits and demerits of Sampling 4.12 Sampling Techniques or Methods 4.13 Probability Sampling Methods		

4.14 Non-Probability Sampling Methods 4.15 Sample Design and Choice of Sampling Technique		
Unit 5	**Data Collection, Processing and Analysis of Data**	**10**
5.1 Collection of Primary Data 5.2 Method of data Collections - Observation, Interview, Questionnaires and Schedules 5.3 Difference between Questionnaires and Schedules 5.4 Some Other Methods of Data Collection 5.5 Collection of Secondary Data 5.6 Selection of Appropriate Method for Data Collection 5.7 Case Study Method 5.8 Processing Operations and Some Problems in Processing 5.9 Elements/Types of Data Analysis 5.10 Statistics in Research 5.11 Measures of Central Tendency, Dispersion, Asymmetry (Skewness) 5.12 Measures of Relationship - Chi-Square, t-test, ANNOVA(f-test),Z-test 5.13 Simple Regression Analysis, and Multiple Correlation and Regression 5.14 Partial Correlation and Association in Case of Attributes 5.15 Quantitative and Qualitative Data Analysis Tools		
Unit 6	**Interpretation and Report Writing**	**8**
6.1 Meaning of Interpretation, Why Interpretation? 6.2 Technique of Interpretation 6.3 Precaution in Interpretation 6.4 Significance of Report Writing 6.5 Different Steps in Writing Report 6.6 Layout of the Research Report 6.7 Types of Reports (Research Proposal/Synopsis, Research Paper, and Thesis) 6.8 Oral Presentation 6.9 Mechanics of Writing a Research Report 6.10 Precautions for Writing Research Reports		
Unit 7	**Publication Ethics and Open Access Publishing**	**8**
7.1 Publication ethics: definition, introduction and importance 7.2 Best practices/standards setting initiatives and guidelines: COPE, WAME, etc. 7.3 Conflicts of interest 7.4 Publication misconduct: definition, concept, problems that lead to unethical behavior and vice versa, types 7.5 Violation of publication ethics, authorship and contributorship 7.6 Identification of publication misconduct, complaints and appeals 7.7 Predatory publishers and journal 7.8 Open access publications and initiatives 7.9 SHERPA/RoMEO online resource to check publisher copyright & self-archiving		

policies

7.10 Software tool to identify predatory publications developed by SPPU

7.11 Journal finder/ journal suggestion tools viz. JANE, Elsevier Journal Finder, Springer Journal Suggester, etc.

7.12 E-Resources for research: Google Scholar, Shodh Ganaga, ShodhGangotri

Learning Outcomes

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

Learning Resources

1. Researching Information Systems and Computing by Briony J Oates, SAGE SOUTH ASIA Ed 2.
2. Research Methodology: A Step-by-Step Guide for Beginners, Kumar, Pearson Education.
3. Research Methodology Methods and Techniques, Kothari, C. R., Wiley Eastern Ltd.
4. The Research Methods Knowledge Base, by William M. K. Trochim, James P. Donnelly
5. Introducing Research Methodology: A Beginner‟s Guide to Doing a Research Project, Uwe Flick
6. A Guide to Research and Publication Ethics by Partha Pratim Ray, New Delhi Publishers
7. RESEARCH & PUBLICATION ETHICS by Wakil kumar Yadav, NOTION PRESS
8. Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd.

Semester II

Course Code: PCSMJ121
Title : Web Frameworks

Teaching Scheme	No. of Credits	Examination Scheme
60 lectures	4	CE : 30 marks ESE: 70 marks

Prerequisites

1. Basic knowledge of JavaScript.
2. Basics of web application development.
3. Basic understanding of Client and Server programming.

Learning Objectives

5. To introduce students to modern web technologies.
6. To learn and use server side programming using Node.js.
7. To learn how to develop web applications using Express.JS.
8. To learn how to develop web applications using Django Python Web Framework.

Course Contents

Unit 1	Java Script Basics	4

1.15 Java Script data types
1.16 Variables, Functions, Events, Regular Expressions
1.17 Array and Objects in Java Script
1.18 Java Script HTML DOM
1.19 Promises and Callbacks

Unit 2	Introduction to Node JS	3

2.8 Introduction
2.9 What is Node JS and its advantages
2.10 Traditional Web Server Model
2.11 Node JS Process model
2.12 Installation of Node JS
2.13 Node JS event loop

Unit 3	Node JS Modules	4

3.6 Functions
3.7 Buffer
3.8 Module and Module Types
3.9 Core Module, Local Module
3.10 Directories as module
3.11 Module.exports

Unit 4	Node Package Manager	3

4.1 What is NPM?
4.2 Installing package locally
4.3 Adding dependencies in package.json
4.4 Installing packages globally
4.5 Updating packages
4.6 Managing Dependencies

Unit 5	**Web Server**	**3**
5.1 Creating Web Server 5.2 Handling HTTP requests 5.3 Sending Requests 5.4 HTTP Streaming		
Unit 6	**File System**	**4**
6.1 FS Model 6.2 Files and Directories 6.3 Streams 6.4 Reading and Writing Files 6.5 Reading and Writing Directories 6.6 Other File Operations		
Unit 7	**Events**	**4**
7.1 Asynchronous JS 7.2 Asynchronous control flow with callbacks 7.3 Promises 7.4 EventEmitter Class 7.5 ASync/Await 7.6 Returning Event Emitter 7.7 Inheriting Events		
Unit 8	**Working with Databases**	**7**
8.2 Connection String 8.3 Configuring 8.4 Working with Select command 8.5 Various database operations 8.6 MongoDB 8.7 Mongoose ODM 8.8 Mongoose Schema 8.9 Mongoose Model 8.10 Querying with Mongoose		
Unit 9	**Express JS**	**14**
9.1 Introduction to Express JS 9.2 Express JS Installation 9.3 Processing GET and POST in Express JS 9.4 Express JS Requests and Responses 9.5 Express JS Rendering 9.6 Express JS Routing 9.7 Template Engines in Express JS 9.8 REST API 9.9 Cookies and Sessions in Express JS 9.10 Error Handling in Express JS		
Unit 10	**Introduction to DJango**	**14**
10.1 What is DJango 10.2 DJango and Python		

10.3 DJango MVT
10.4 How to get and install DJango
10.5 About 3 core files : model.py, urls.py, views.py
10.6 Creating DJango project and app
10.7 Models in DJango
10.8 Views in DJango
10.9 Templates in DJango
10.10 DJango Forms
10.11 DJango Form Validation
10.12 DJango File Uploads
10.13 Setting up database connections
10.14 DJango CRUD applications

Learning Outcomes

1. Students will be ready with the technology which is used widely in Industry as a part of full stack developer.
2. Students will know the powerful way to develop the web application in Python.
3. Students will understand asynchronous programming.
4. Students will understand the development of web applications using Express.JS.
5. Build and deploy robust Django Web App.

Learning Resources

1. Node.js complete reference guid , velentin Bojinov, David Herron, Dioge Resende, packt Publishing Ltd
2. Mastering Nod.js By Sandro Pasquali , packt Publishing
3. Smashing Node.js, Java Script Everywhere , Guillermo Rauch, John wiley & Sons
4. Django for Beginners : Build websites with Python and Django Kindle Edition by William S Vincent
5. Two Scoops of Django 1.11 : Best Practices for the Django web Framework Book by Danial Roy Greenfeld and Audrey Roy Greenfeld
6. Web Development with Django CookBook by Aidas Bendoraits Second Edition PACKT Publishing

Course Code: PCSMJ122
Title: Artificial Intelligence and Machine Learning

Teaching Scheme	No. of Credits	Examination Scheme
60 lectures	4	CE: 30 marks ESE: 70 marks

Prerequisites

1. Concepts of Data structures, Design and Analysis of algorithms.
2. Strong data analytics skills
3. Programming in Python (NumPy, SciPy, Pandas, Matplotlib, Seaborn, SciKit-Learn, StatsModel)
4. Familiarity with Probability Theory, Multivariable Calculus, Linear Algebra

Course 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 introduce students to the basic concepts and techniques of Machine Learning.
4. To write Python programs using machine learning algorithms for solving practical problems.
5. To understand about Machine Learning Library and use cases.

Course Contents

Chapter 1	**Problem Solving**	**12**

1.1 Introduction to AI
1.2 AI Applications
1.3 Defining AI problems as state space search
1.4 Problem-solving agents search algorithms
1.5 Uninformed search strategies
1.6 Heuristic search strategies
1.7 Local search and optimization problems
1.8 Adversarial search
1.9 Constraint satisfaction problems (CSP)

Chapter 2	**Knowledge Representation**	**12**

2.1 Representations and Mappings
2.2 Approaches to Knowledge Representation
2.3 Knowledge representation method
2.4 Propositional Logic
2.5 Predicate logic
2.6 Representing Simple Facts in Logic Resolution, forward and backward chaining
2.7 Game Playing- Minimax Search Procedures, Adding alpha-beta cutoffs

Chapter 3	**Introduction to Machine Learning**	**12**

3.1 Difference between Data Science, Artificial Intelligence and Machine Learning and 3.2 Deep Learning 3.3 Why Learn and What is Learning, What is Machine Learning 3.4 Traditional Programming Vs. Machine Learning, Machine Learning Process 3.6 Types of Data: Discrete, Continuous, nominal, ordinal 3.6 Descriptive and Inferential Statistics: Probability, Distribution, Distance Measures (Euclidean and Manhattan), Correlation and Regression, Hypothesis Testing 3.7 Data Preprocessing: Creating our own dataset, Importing the dataset, Handling 3.8 Missing Data, Splitting the dataset into the Training set and Test set, Feature Scaling		
Chapter 4	**Machine Learning Models**	**5**
4.1 Type of Learning- Supervised, Unsupervised, Reinforcement and Semi-Supervised Learning 4.2 Components of Generalization Error (Bias, Variance, Underfitting, Over fitting) 4.3 A Learning System Cycle and Design Cycle		
Chapter 5	**Supervised Learning Models**	**10**
5.1 Linear Regression - Simple, Multiple, Polynomial 5.2 Non-linear Regression – Decision Tree, Support Vector 5.3 Evaluating Regression Models: R-squared, Adjusted R-squared 5.4 K – Nearest Neighbours (KNN) classifier 5.5 Logistic Regression 5.6 Naive Bayes Classification 5.7 Support Vector Machine, Kernel SVM 5.8 Decision Tree Classification 5.9 Ensemble Learning, Entropy, Information Gain 5.10 Random Forest Classification 5.11 Evaluating Classification Models		
Chapter 6	**Unsupervised Learning Models**	**6**
6.1 Types of Clustering: Hierarchical, Partitioning 6.2 K-means Clustering: Within Clusters Sum of Squares (WCSS), Selecting the optimal number of clusters by Elbow Method, Agglomerative Clustering: Dendrogram 6.3 Key Terms in Association Rule mining: Support, Confidence, and Lift Association Rule mining algorithms: Apriori, Eclat, FP Growth		
Chapter 7	**Reinforcement Learning**	**3**
7.1 The multi arm Bandit problem 7.2 Upper Confidence Bound, 7.3 Thompson Sampling 7.4 UCB Vs Thompson sampling		
Learning Outcomes 1. Understand the algorithms based on Artificial Intelligence that make it useful for real-world problems. 2. Design simple applications using Artificial Intelligence. 3. Recognize the characteristics of machine learning that make it useful to real-world problems. 4. Process available data using python libraries and predict outcomes using Machine		

Learning algorithms to solve a given problem.

5. Able to estimate Machine Learning models efficiency using suitable metrics.

Learning Resources:

1. Artificial Intelligence with Python, PrateekJoshi Packt Publishing Ltd, First edition(2017)
2. Machine learning course material by Andrew Ng, Stanford university
3. Data Mining: Practical machine learning tools and techniques. Witten, Ian H, and Eibe Frank, Morgan Kaufmann, 2005.

Course Code: PCSMJ123
Title : Laboratory Course on Web Frameworks

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs/ week per batch	2	CE : 15 marks ESE: 35 marks

Prerequisites

1. Basics of Node.JS.
2. Basics of Express.JS and DJango frameworks.

Learning Objectives

1. To understand the usage of Node.JS technology along with various web development frameworks i.e. ExpressJS and DJango.

Assignments

1. Create an HTML form that contain the Student Registration details and write a JavaScript to validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50.
2. Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.
3. Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.
4. Create a Simple Web Server using node js.
5. Write node js script to build Your Own Node.js Module. Use require ('http') module is a built-in Node module that invokes the functionality of the HTTP library to create a local server. Also use the export statement to make functions in your module available externally. Create a new text file to contain the functions in your module called, "modules.js" and add this function to return today's date and time.
6. Using nodejs create a web page to read two file names from user and append contents of first file into second file
7. Create a Node.js file that writes an HTML form, with an upload field.
8. Create a js file named main.js for event-driven application. There should be a main loop that listens for events, and then triggers a callback function when one of those
   Events is detected.
9. Create a Node.js file that demonstrates creating databases, collections and documents in MongoDB.
10. Create a node.js file that Select all records from the "customers" table, and display the result object on console.
11. Create a node.js file that Insert Multiple Records in "student" table, and display the result object on console
12. Create a node.js file that Select all records from the "customers" table, and delete the specified record.
13. Using node js create a User Login System.
14. Using node js create a Recipe Book.
15. Write node js application that transfer a file as an attachment on web and enables browser to prompt the user to download file using express js.

16. Create your Django app in which after running the server, you should see on the browser, the text “Hello! I am learning Django”, which you defined in the index view.
17. Design a Django application that adds web pages with views and templates.
18. Develop a basic poll application (app).It should consist of two parts: a) A public site in which user can pick their favorite programming language and vote. b) An admin site that lets you add, change and delete programming languages.
19. Implement Login System using Django.
20. Create your own blog using Django.

Learning Outcomes

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

1. Understand the development of web application architecture leading to a more modular approach.
2. Sketch out the components that would be used in a range of approaches to web application architecture

Course Code: PCSMJ124
Title: Laboratory Course on Artificial Intelligence and Machine Learning

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs/ week per batch	2	CE : 15 marks ESE: 35 marks

Prerequisites:

1. Basics of Python, Python statements
2. Methods & Functions using Python
3. Concepts on Artificial Intelligence: Knowledge Representation, Searching algorithms

Course Objectives:

1. To write Python programs using Artificial Intelligence algorithms for solving practical problems.

Assignments :

1. Write a Python program to implement Simple Chatbot.
2. Write a Python program to implement Breadth First Search Traversal and Depth First Search Traversal.
3. Write a Python program to implement a Water Jug Problem.
4. Write a Python program to Prepare Scatter Plot (Use Iris Dataset)
5. Write a Python program to find all null values in a given data set and remove them.
6. Write a Python program the Categorical values in numeric format for a given dataset.
7. Write a Python program to implement simple Linear Regression for predicting house prices.
8. Write a Python program to implement multiple Linear Regression for a given dataset.
9. Write a Python program to implement Polynomial Regression for a given dataset.
10. Write a Python program to Implement Naïve Bayes.
11. Write a Python program to Implement Decision Tree on whether or not to play tennis.
12. Write a Python program to implement linear SVM.
13. Write a Python program to implement k-nearest Neighbors ML algorithm to build prediction model (Use Forge Dataset)
14. Write a Python program to implement k-means algorithm on a synthetic dataset.
15. Write a Python program to implement Agglomerative clustering on a synthetic dataset.

Learning Outcomes:

1. Understand Artificial Intelligence algorithm implementations for solving real-world problems.
2. Understand the designing applications using Artificial Intelligence.

Course Code: PCSMJ125A
Title : DOT NET

Teaching Scheme 30 lectures	No. of Credits 2	Examination Scheme CE : 15 marks ESE: 35 marks

Prerequisite

1. Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
2. Familiarity with programming languages such as C++ and/or Java.

Learning Objectives:

1. To understand the DOTNET framework
2. Develop deep understanding of C# language features
3. Build strong concepts of OOP's and implement the same in C#.
4. To understand the concept of multi-threading & files
5. To understand and implement the controls & properties of Windows forms
6. To develop database centric applications using ADO.NET

Course Contents

Unit 1	Introduction to .Net FrameworkIntroduction to .Net Framework	22

1.1 Overview of .NET framework &.Net Architecture
1.2 The Common Language Runtime (CLR)
1.3 Microsoft Intermediate Language (MSIL) Code
1.4 Just In Time Compilers (JITers)
1.5 The Framework Class Library (FCL)
1.6 The Common Languages Specification (CLS)
1.7 The Common Type System (CTS)
1.8 Garbage Collection (GC)Overview of .NET framework &.Net Architecture
1.9 The Common Language Runtime (CLR)
1.10 Microsoft Intermediate Language (MSIL) Code
1.11 Just In Time Compilers (JITers)
1.12 The Framework Class Library (FCL)
1.13 The Common Languages Specification (CLS)
1.14 The Common Type System (CTS)
1.15 Garbage Collection (GC)

Unit 2	Introduction to C#.NetIntroduction to C#.Net	44

2.1 Basics of C#. Language (Console Application)
2.2 Namespace, Variables and Expressions
2.3 Type Conversion
2.4 Boxing and Un-boxing
2.5 Flow Control
2.6 Functions
2.7 Debugging and error handling
2.8 Array - One-dimensional & two-dimensional array

2.1 Exception handling- System Defined and User DefinedBasics of C#. Language (Console Application) 2.2 Namespace, Variables and Expressions, 2.3 Type Conversion 2.4 Boxing and Unboxing 2.5 Flow Control 2.6 Functions 2.7 Debugging and error handling 2.8 Array - One-dimensional & two-dimensional array 2.9 Exception handling - System Defined and User Defined		
Unit 3	**OOPS with C#OOPS with C#**	**57**
3.1 Object Oriented Concept 3.2 Object and Classes 3.3 Class properties: Access modifiers, Implementation of class 3.4 Constructor 3.5 Inheritance 3.6 Polymorphism & Interface 3.7 Abstract Class 3.8 Delegates 3.9 Multicasting & Anonymous Methods 3.10 Data Structure : ArrayList , Collection , Dictionary , Hash Table 3.11 Multithreading IO Stream : Stream Reader, Stream Writer , File Mode , Opening & Closing File , Random Access File 3.12 Assembly Components : .NET Assembly features , Structure of Assemblies , Calling assemblies, private and shared assemblies 3.1 Object Oriented Concept 3.2 Object and Classes 3.3 Class properties: Access modifiers, Implementation of class 3.4 Constructor 3.5 Inheritance 3.6 Polymorphism & Interface 3.7 Abstract Class 3.8 Delegates 3.9 Multicasting & Anonymous Methods 3.13 Data Structure - ArrayList , Collection , Dictionary , Hash Table		
Unit 4	**Windows ProgrammingMultithreading I/O Stream and Assembly Components**	**6**
4.1 Windows Forms Menus and Tool Bars, SDI and MDI applications, Building MDI applications. 4.2 Basic Controls Button, TextBox, Label, RadioButton, CheckBoxDateTimePicker, Timer ,PictureBox, ComboBox, ListBox, RichTextBox, MonthCalender 1. Container & Dialog Control GroupBox, Panel, Common Dialog boxes, ProgressBarStream Reader, Stream Writer 2. File Mode 3. Opening & Closing File 4. Random Access File 5. .NET Assembly features 6. Structure of Assemblies 7. Calling assemblies, private and shared assemblies		

Unit 5	**Database Connectivity using ADO.NetWindows Programming and Database Connectivity using ADO.NET**	**612**

5.1 ADO.NET Architecture
5.2 Connection object, Commend Object
5.3 Dataset, DataReader & DataAdapter
5.4 SQL Commands (Insert, Delete,Update,Select)
5.5 Accessing Data with ADO.NET

1. Datagridview Data Binding: Insert, Update, Delete recordsWindows Forms Menus and ToolBars, SDI and MDI applications, Building MDI applications.
2. Basic Controls Button, TextBox, Label, RadioButton, CheckBoxDateTimePicker, Timer ,PictureBox, ComboBox, ListBox, RichTextBox, MonthCalender
3. Container & Dialog Control GroupBox, Panel, Common Dialog boxes, ProgressBar
4. ADO.NET Architecture
5. Connection object, Commend Object
6. Dataset, DataReader & DataAdapter
7. SQL Commands (Insert, Delete,Update,Select)
8. Accessing Data with ADO.NET
9. Datagridview Data Binding: Insert, Update, Delete records

Learning Outcomes

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

1. Understand the features of Dot Net Framework along with the features of C#
2. Interpret and Develop Interfaces for real-time applications.
3. Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.
4. Design & Implement the application using multithreading & File handling
5. Design and Implement Windows Application using Windows Forms & tools application using Database in C#

Design and Implement Custom Application Using Windows Form & ADO.NET in C#On Completion of this course, student will be able to -

1. Understand the features of Dot Net Framework along with the features of C#
2. Interpret and Develop Interfaces for real-time applications.
3. Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.
4. Design & Implement the application using multithreading & File handling
5. Design and Implement Windows Application using Windows Forms & tools application using Database in C#
6. Design and Implement Custom Application Using Windows Form & ADO.NET in C#

Learning Resources

1. Programming in C#, E.Balagurusamy,
2. Professional C# ,Wrox Publication
3. C# The Complete Reference", Shildt, TMH

1. Database Programming with C#, By Carsten Thomsen, ApressProgramming in C#, E.Balagurusamy,
2. Professional C# ,Wrox Publication
3. C# The Complete Reference", Shildt, TMH
4. Database Programming with C#, By Carsten Thomsen, Apress

Course Code: PCSMJ125B
Title : Laboratory course on DOT Net

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs/ week per batch	2	CE : 15 marks ESE: 35 marks

Prerequisite

1. Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
2. Familiarity with programming language such as C++ and/or Java.

Learning Objectives:

1. To understand the DOTNET framework
2. Develop deep understanding of C# language features
3. Build strong concepts of OOP's and implement the same in C#.
4. To understand the concept of multi-threading & files
5. To understand and implement the controls & properties of Windows forms
6. To Develop database centric applications using ADO.NET.

Assignments :

1. Assignment on Basic C# programAssignment on OOPs concept.
2. Assignment on Data structure
3. Assignment on Data Structure
4. Assignment on multithreading
5. Assignment on Assembly and exception handling
6. Assignment on Windows programming and database connectivityAssignment on basic C# program and OOPs concept
7. Assignment on Data Structure , Multithreading and IO Stream
8. Assignment on Assembly and Exception Handling
9. Assignment on Windows Programming
10. Assignment on Database Connectivity using ADO.Net

Learning Outcomes:

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

1. Understand the features of Dot Net Framework along with the features of C#
2. Interpret and Develop Interfaces for real-time applications.
3. Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.
4. Design & Implement the application using multithreading & File handling
5. Design and Implement Windows Application using Windows Forms & tools application using Database in C#

Design and Implement Custom Application Using Windows Form & ADO.NET in C#On Completion of this course, student will be able to -

1. Understand the features of Dot Net Framework along with the features of C#
2. Interpret and Develop Interfaces for real-time applications.
3. Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.

4. Design & Implement the application using multithreading & File handling
5. Design and Implement Windows Application using Windows Forms & tools application using Database in C#
6. Design and Implement Custom Application Using Windows Form & ADO.NET in C#
6.

Course Code: PCSMJ126A
Title : Software Project Management

Teaching Scheme 30 Lectures	No. of Credits 2	Examination Scheme CE : 15 marks SEE: 35 marks

Prerequisites

Student should have basic knowledge of Software Engineering concepts.

Course Objectives

1. This course is aimed to understand the growing need for better project management.
2. To understand key elements of the project management framework including project stakeholders, the project management knowledge areas, common tools and techniques, and project success.
3. Software Project Management covers skills that are required to ensure successful development of medium and large scale software projects.

Course Contents

Unit 1	**Introduction to Project Management**	**4**

1.1 What is a Project?
1.2 What is Project management?
1.3 Project phases and project life cycle
1.4 Organizational structure
1.5 Qualities of Project Manager

Unit 2	**Project Management Components**	**6**

2.1 Project Integration Management-Project plan development and execution
2.2 Change controls
2.3 Change Control Board(CCB)
2.4 Configuration management

Unit 3	**Project Management-Core Knowledge Areas**	**12**

3.1 Scope Management
3.1.1 Strategic planning
3.1.2 Scope planning,definition
3.1.3 Scope verification,control
3.2 Time Management
3.2.1 Activity planning
3.2.2 Schedule development and control
3.2.3 Gantt Chart
3.3 Cost Management
3.3.1 Cost estimation and control
3.3.2 COCOMO model
3.4 Quality Management
3.4.1 Quality planning
3.4.2 Quality assurance

Unit 4	**Project Management-Facilitating Knowledge Areas**	**8**

4.1 Human Resource Management
4.1.1 Organizational planning
4.1.2 Staff acquisition
4.2 Communication Management
4.2.1 Information distribution
4.2.2 Reporting
4.3 Risk Management
4.3.1 Risk identification
4.3.2 Risk Quantification and control
4.4 Procurement Management
4.4.1 Solicitation management and control
4.4.2 Contract administration

Learning Outcomes

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

1. Understand the organizational structure and to manage the projects in the industry.
2. Understand the skills required to become a good project manager.
3. Select and apply tools and techniques in carrying out work of all knowledge areas of project management.

Learning Resources

1. Information Technology Project Management,Kathy Schwalbe,cengage publication,Revised 6th edition.
2. Software Engineering,Roger Pressman,McGraw Hill,6th edition,.
3. Software Metrics for Project Management and process improvement,Robert B. Grady,Prentice Hall.

Course Code: PCSMJ126B
Title : Laboratory Course on Software Project Management (project management tool)

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs/week per batch	2	CE : 15 marks SEE: 35 marks

Prerequisites

1. Students should have a basic knowledge of software development.
2. Be familiar with various knowledge areas of software project management.

Course Objectives

1. To understand the role and importance of software management tools in effectively managing software projects.
2. To learn how these tools enhance project planning, tracking, collaboration and communication within project team.

Assignments

Assignment 1:
Consider a project for developing a responsive e-commerce website. Use SPM tools and techniques for the given project to prepare and manage a detailed project plan, decide the scope and create a board to assign tasks, design the project schedule with milestones and track the progress of the project. Collaborate effectively with team members and keep a track of cost of project.

Assignment 2:
Consider a project for developing a mobile banking application for a financial institution. Use SPM tools and techniques for the given project to prepare and manage a detailed project plan, decide the scope and create a board to assign tasks, design the project schedule with milestones and track the progress of the project. Collaborate effectively with team members and keep a track of cost of project.

Assignment 3:
Consider a project for developing a patient management system. Use SPM tools and techniques for the given project to prepare and manage a detailed project plan, decide the scope and create a board to assign tasks, design the project schedule with milestones and track the progress of the project. Collaborate effectively with team members and keep a track of cost of project.

Assignment 4:
Consider a project for developing a student information system. Use SPM tools and techniques for the given project to prepare and manage a detailed project plan, decide the scope and create a board to assign tasks, design the project schedule with milestones and track the progress of the project. Collaborate effectively with team members and keep a track of cost of project.

Learning Outcomes:
On completion of the course, student will be able to:
CO1- gain practical experience and proficiency in using software management tools commonly used in industry.
CO2- navigate the tools, create and manage project tasks, assign resources, track progress and generate reports.
CO3- Apply project management tools and techniques to an IT project.