

**Haribhai V. Desai College
of Arts, Science & Commerce, Pune.
(Autonomous)**

Faculty of Science and Technology

B.Sc. (Computer Science) Program



**Syllabus For
T.Y B.Sc. (Computer Science)**

Choice Based Credit System (CBCS)

Syllabus Under National Education Policy (NEP)

with effect from Academic Year 2026-27

Title of the Program: B.Sc. (Computer Science)

Preamble:

The B. Sc. (Computer Science) and B. Sc. (Computer Science) (Honors) and (Research) course is a systematically designed program with Computer Science as a major subject under the faculty of Science and Technology. The objective of the course is to prepare students to undertake careers involving problem solving using computer science and technologies, or to pursue advanced studies and research in computer science. The syllabus which comprises of Computer Science (Major) subject along with that of the three allied subjects (Mathematics, Electronics and Statistics) (Minor) covers the foundational aspects of computing sciences and also develops the requisite professional skills and problem-solving abilities using computing sciences.

Introduction:

At the first year of under-graduation, the basic foundations of two important skills required for software development are laid. A course in problem solving and programming along with a course in database fundamentals forms the preliminary skill set for solving computational problems. The practical courses are designed to supplement the theoretical training in the year. Along with Computer Science (Major), VSC and SEC courses help in building a strong technical foundation. Another aspect of this course is IKS which tells about the rich heritage and advancement of India in the field of computation.

In the second year of under-graduation, computational problem-solving skills are further strengthened by a course in Data structures, C++ and python programming. Software engineering concepts that are required for project design are also introduced. Essential concepts of computer networking are also introduced this year. The practical course included in both semesters complements the theory courses. Field projects/ OJT is introduced so that students can implement the concept they have learnt in first year.

In Second Year, the “Subject 1 : Computer Science” will be the Major Subject and the Minor subject will be chosen from “Subject 2 or Subject 3”. Subject 2 and Subject 3 will not be available as Major Subjects in Second Year and Third Year.

At the third year of under-graduation, all the subjects are designed to fulfill core Computer Science requirements as well as meet the needs of the software industry. Theory courses are adequately supplemented by hands-on practical courses. Major elective courses are taking care of recent advancement in the field of computer science. Minor and Skill Enhancement courses enable the students to acquire additional skills.

At the fourth year (honors) and (research) of under-graduation, all the subjects are designed to fulfill core Computer Science requirements as well as meet the needs of the software industry. Practical courses and field projects enable students to get hands-on training. Various learning tracks are open through Major elective courses. Research methodology course will create interest among the students to carry research in the field of computer science.

Objectives:

- To develop problem solving abilities using a computer.
- To build the necessary skillset and analytical abilities for developing computer-based solutions for real life problems.
- To train students in professional skills related to the Software Industry.
- To prepare the necessary knowledge base for research and development in Computer Science.
- To help student's build-up a successful career in Computer Science and to produce entrepreneurs who can
- Innovate and develop software products.

Eligibility:

1. H.S.C. (10 + 2) Science stream with Mathematics.
2. Three years diploma course after S.S.C. (10th std.) of Board of Technical Education conducted by Government of Maharashtra or its equivalent.

Program Outcomes:

PO No.	Outcomes
PO1	Develop creative skills, critical thinking, analytical skills and research to address the real- world problems using computational skills.
PO2	Understand and apply mathematical foundation, computing and domain knowledge and develop computing models for defined problems.
PO3	Understand software project management and computing principles with computing knowledge to manage projects in multidisciplinary environments.
PO4	Illustrate the concepts of systems fundamentals, including architectures and organization, operating systems, networking and communication.
PO5	Understand and apply the concepts of Digital Electronics, Computer Architecture, IoT etc.
PO6	Recognize the need for and develop the ability to engage in continuous learning as a Computing professional.
PO7	Apply modern computing tools, skills and techniques necessary for innovative software solutions.
PO8	Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations.
PO9	Gain Self Discipline and commit Professional Ethics in global economic environment.
PO10	Individual & Team Work: Ability to work as a member or leader in diverse teams in multidisciplinary environment.
PO11	Identify opportunities, entrepreneurship vision and use innovative ideas to create value and wealth for the betterment of the individual and society.

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
 (Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

SEMESTER-I

Level: 4.5 (First Year)

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hour/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CIE	SEE	Total
Subject-1	CS-101-TH	Problem Solving using 'C' Programming	2		2		20	30	50
	CS-102-PR	Lab Course based on CS-101-TH		2		4	20	30	50
Subject-2	ELC-101-TH	Principles of Analog Electronics	2		2		20	30	50
	ELC-102-PR	Electronics Practical Course I		2		4	20	30	50
Subject-3	MTC-101-TH	Matrix Algebra	2		2		20	30	50
	MTC-102-PR	Mathematics Practical I		2		4	20	30	50
IKS (2)	IKS-101-TH	Introduction to Indian Knowledge System (Generic)	2		2		20	30	50
GE/OE * (2)	OE-101-CS - TH/ OE-102-CS - TH/ OE-103-CS-TH	Fundamentals of Computers I / Office Automation I / Introduction to Google Apps I / (Offering to other stream students)	2		2		20	30	50
SEC (2)	SEC-101-STS-PR	Statistical Methods for Computer Science I		2		4	20	30	50
AEC (2)	AEC-104-ENG-TH	Professional Communication Skills-I	2		2		20	30	50
VEC (2)	VEC-101-ENV-TH	Environmental Education-I	2		2		20	30	50
TOTAL			14	08	14	16	165	385	550

* The subjects offered to other faculty students under OE vertical are OE-101-CS-TH/ OE-102-CS-TH/OE-103-CS- TH
 The students of B.Sc. (Computer Science) will opt the subjects offered by other faculty given in College Basket.

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

SEMESTER-II

Level: 4.5 (First Year)

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hour/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CIE	SEE	Total
Subject-1	CS-151-TH	Advanced C Programming	2		2		20	30	50
	CS-152-PR	Lab Course Based on CS-151- TH		2		4	20	30	50
Subject-2	ELC-151-TH	Principles of Digital Electronics	2		2		20	30	50
	ELC-152-PR	Electronics Practical Course II		2		4	20	30	50
Subject-3	MTC-151- TH	Graph Theory	2		2		20	30	50
	MTC-152-PR	Mathematics Practical II		2		4	20	30	50
GE/OE* (2)	OE-151-CS - PR/ OE-152-CS - PR/ OE-153-CS- PR	Fundamentals of Computers II / Office Automation II / Introduction to Google Apps II / (Offering to other stream students)		2		4	20	30	50
SEC (2)	SEC-151-STS-PR	Statistical Methods for Computer Science II		2		4	20	30	50
AEC (2)	AEC-154-ENG-TH	Professional Communication Skills-II	2		2		20	30	50
VEC (2)	VEC-151-ENV-TH	Environmental Education-II	2		2		20	30	50
CC (2)	CC	From College Basket		2		4	20	30	50
TOTAL			10	12	10	24	165	385	550

* The subjects offered to other faculty students under OE vertical are OE-101-CS-PR/ OE-102-CS-PR/OE-103-CS- PR The students of B.Sc. (Computer Science) will opt the subjects offered by other faculty given in College Basket.

Exit Option:

Award of UG Certificate in Major with 44 credits and an additional 4 credits core as per university guidelines OR Continue with Major and Minor Continue option: Student will select one subject among the (subject 2 and subject 3) as minor and subject 1 will be major subject.

In Second Year, the “Subject 1: Computer Science” will be Major Subject and the Minor subject will be chosen from “Subject 2 or Subject 3”. Subject 2 and Subject 3 will not be available as Major Subjects in Second Year and Third Year.

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
 (Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

SEMESTER-III

Level- 5.0 (Second Year)

Course Type	Course Code	Course Title	Credit		Teaching Scheme Hr. / Week		Evaluation Scheme & Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (4T+2P)	CS-201-MJ-TH	Data Structure-I	2		2		20	30	50
	CS-202-MJ-TH	Database Management System	2		2		20	30	50
	CS-203-MJ-PR	Lab Course based on CS-201-MJ-TH & CS-202-MJ-TH		2		4	20	30	50
VSC (2T)	CS-221-VSC-TH	Agile & Software Engineering	2		2		20	30	50
FP/OJT/ CEP (2)	CS-231-FP -PR	SEM-III Mini Project		2		4	20	30	50
Minor (2T+2P)	MTC-241-MN-TH OR ELC-241-MN-TH	Group Theory and Applications in Coding OR Microcontroller Architecture and Programming (MAP)	2		2		20	30	50
	MTC-242-MN-PR OR ELC-242-MN-PR	Python Programming Language-I OR Electronics Practical Course-III		2		4	20	30	50
GE/OE (2T)	OE-201-CS-TH OR	Web Designing-I OR	2		2		20	30	50
	OE-202-CS-TH	Digital Banking and Fintech Awareness	2		2		20	30	50
IKS (2T)	IKS-201-CS-TH	Vedik Mathematics in Algorithm Optimization	2		2		20	30	50
AEC (2T)	AEC	From College Basket	2		2		20	30	50
CC (2)	CC	From College Basket	2		2		20	30	50
TOTAL			16	6	16	12	165	385	550

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

SEMESTER-IV

Level- 5.0 (Second Year)

Course Type	Course Code	Course Title	Credit		Teaching Scheme Hr. / Week		Evaluation Scheme & Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (6T+2P)	CS-251-MJ-TH	Data Structure-II	2		2		20	30	50
	CS-252-MJ-TH	Relational Database Management System	2		2		20	30	50
	CS-253-MJ-PR	Lab Course based on CS-251-MJ- TH & CS-252-MJ-TH		2		4	20	30	50
VSC (2T/P)	CS-271-VSC-PR	Python Programming		2		4	20	30	50
FP/OJT/ CEP (2)	CS-281-CEP -PR	SEM-IV Mini Project		2		4	20	30	50
Minor (2T+2P)	MTC-291-MN-TH OR ELC-291-MN-TH	Applied Mathematics OR Embedded Systems and IoT using Raspberry Pi and Arduino	2		2		20	30	50
	MTC-292-MN-PR OR ELC-292-MN-PR	Python Programming Language-II OR Electronics Practical Course–IV		2		4	20	30	50
GE/OE (2P)	OE-251-CS-PR OR	Web Designing-II OR		2		4	20	30	50
	OE-252-CS-TH	Digital Skills for Entrepreneurs	2		2		20	30	50
SEC (2P)	SEC-251-CS-PR OR SEC-252-STS-PR	Computer Networking OR Statistical Analysis using R Programming		2		4	20	30	50
AEC (2T)	AEC	From College Basket	2		2		20	30	50
CC (2)	CC	From College Basket	2		2		20	30	50
TOTAL			10	12	10	24	165	385	550

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

SEMESTER-V

Level- 5.5 (Third Year)

Course Type	Course Code	Course Title	Credit		Teaching Scheme Hr. / Week		Evaluation Scheme & Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Mandatory (8T+4P)	CS-301-MJ-TH	Core Java	2		2		20	30	50
	CS-302-MJ-TH	Web-Technology-I	2		2		20	30	50
	CS-303-MJ-TH	Operating Systems	2		2		20	30	50
	CS-304-MJ-TH	Theory of Computer Science	2		2		20	30	50
	CS-305-MJ-PR	Lab Course based on CS-301-MJ-TH		2		4	20	30	50
	CS-306-MJ-PR	Lab Course based on CS-302-MJ-TH		2		4	20	30	50
Major Elective (2T+2P)	CS-310-MJ-TH	Data Science	2		2		20	30	50
	CS-311-MJ-PR	Lab Course based on CS-310-MJ-TH		2		4	20	30	50
	OR								
	CS-312-MJ-TH	Database Technologies	2		2		20	30	50
	CS-313-MJ-PR	Lab Course based on CS-312-MJ-TH		2		4	20	30	50
VSC (2T/P)	CS-321-VSC-PR	Lab Course based on CS-303-MJ-TH		2		4	20	30	50
FP/OJT/CEP (2FP/CEP)	CS-331-FP-PR	Project		2		4	20	30	50
Minor (2T)	ELC-341-MN- TH	Digital Communication and Networking	2		2		20	30	50
	OR MTC-341-MN- TH	OR Optimization Techniques							
TOTAL			12	10	12	20			550

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

SEMESTER-VI

Level- 5.5 (Third Year)

Course Type	Course Code	Course Title	Credit		Teaching Scheme Hr. / Week		Evaluation Scheme & Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Mandatory (8T+4P)	CS-351-MJ-TH	Advanced Java	2		2		20	30	50
	CS-352-MJ-TH	Web Technology-II	2		2		20	30	50
	CS-353-MJ-TH	Design Framework	2		2		20	30	50
	CS-354-MJ-TH	Compiler Construction	2		2		20	30	50
	CS-355-MJ-PR	Lab Course based on CS-351- MJ- TH		2		4	20	30	50
	CS-356-MJ-PR	Lab Course based on CS-352- MJ- TH		2		4	20	30	50
Major Elective (2T+2P)	CS-360-MJ-TH	Android Programming	2		2		20	30	50
	CS-361-MJ-PR	Lab Course based on CS-360- MJ- TH		2		4	20	30	50
	OR								
	CS-362-MJ-TH	Software Testing Tools	2		2		20	30	50
	CS-363-MJ-PR	Lab Course based on CS- 362- MJ- TH		2		4	20	30	50
VSC(2T/P)	CS-371-VSC-PR	Lab Course based on CS-353- MJ- TH		2		4	20	30	50
FP/OJT/ CEP (4 OJT)	CS-381-OJT-PR	On Job Training		4		8	40	60	100
TOTAL			12	10	12	20			550

Exit option:

Award of UG Diploma in Major and Minor with **88 credits and an additional 4 credits** core as per college guidelines OR Continue with Major and Minor.

ATKT Rules:

Minimum number of credits required to take admission to T.Y. B.Sc.(Computer Science) is 44 credits to be completed from F.Y. B.Sc.(Computer Science) and at least 22 credits from S.Y. B.Sc.(Computer Science)

Assessment and Evaluation:

Scheme of Examination:	Exam Format	Minimum Passing Marks
CIE (20 Marks)	i) Internal Test and ii) Any of the following- A. Assignment B. Seminar C. Presentation D. Oral etc.	Min. 8 Marks (40% Passing)
SEE (30 Marks)	Theory/Practical Examination	Min. 12 Marks (40% Passing)
Total 50 Marks		Min.20 Marks (40% Passing)

SEM-V

Syllabus

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-301-MJ-TH	Core Java	2	TH	CIE-20 SEE-30
Course Hours: 30 Hours					
Course Objectives:					
1 List the basics of object-oriented programming.					
2. To learn the implementation of object-oriented concepts					
3. To introduce the principles of inheritance and polymorphism and demonstrate how they relate to the design of abstract classes					
4. Implement file handling and exception Handling.					
5. Design GUI-based applications using Swing.					
Course Outcomes: (COs)					
1. Identify classes, objects, class members and relationships for a given problem.					
2. Implement different types of object-oriented concepts.					
3. Handle File and directories.					
4. Handle abnormal termination of a program using exception handling					
5. Use Java APIs for GUI development					
Syllabus:					
Chapter 1: Fundamentals of Java Programming (7 Hours)					
Topics Covered:					
1.1 Introduction to Object Oriented Programming concepts,					
1.2 java Basics					
1.2.1. History of Java,					
1.2.2. features of java,					
1.2.2. java environment – JVM, JRE, JDK.					
1.2.3. Basic syntax – data types, variables, operators and Control flow statements					
1.2.4. Java keywords,					
1.2.5. java Garbage Collection					
1.3 Java Tools – javac, java, jdb, javap, Javadoc, javah, javap and jar.					
1.4 Simple Java program,					
1.5 Structure(anatomy) of Java program,					
1.6 Arrays and its application,					
1.7 accepting input using command line arguments and from console (Using BufferedReader and Scanner class)					
1.8 Simple programs demo.					
1.9 Integrated Development Environments (IDEs)					

Chapter 2: Classes and packages (4 Hours)
Topics Covered: 2.1 Classe- built-in and user define classes, 2.2 Encapsulation - Access Specifies (public, protected, private, default), Access modifiers, 2.3 Array of Objects 2.4 Constructor, Overloading Constructors 2.5 use of ‘this’ and ‘static’ Keyword 2.6 Package- creation, uses, types and access. 2.7 Wrapper Classes, autoboxing/unboxing 2.8 type conversion/casting -Widening casting, Narrowing casting
Chapter 3: Inheritance (4 Hours)
Topics Covered: 3.1 Inheritance -Basics (extends Keyword) and Types of Inheritance, 3.2 Super class and Subclass 3.3 use of Super Keyword 3.4 Polymorphism 3.4.1 Method Overloading (compile time polymorphism/early binding) 3.4.2 Method Overriding (runtime polymorphism/ Dynamic Method Dispatch), 3.5 Use of final, 3.6 Use of abstract class and abstract methods, 3.7 Interfaces 3.7.1 Defining and Implementing Interfaces 3.7.2 Runtime polymorphism using interface 3.7.3 Functional and Marker Interfaces
Chapter 4: String, Streams and File Handling (4 Hours)
Topics Covered: 4.1 String class and String Buffer Class, 4.2 Using the File class 4.3 Stream classes, Byte Stream classes, Character Stream Classes, 4.4 Creation of files, Reading/Writing characters and bytes,
Activities:
Oral
Chapter 5: Exception Handling (4 Hours)
Topics Covered: 5.1 Errors and Exception, Exception class, 5.2 Exception Types-Checked and Unchecked exception, 5.3 Exception handling using try, catch, throw, throws and finally keywords. 5.4 Creating user defined exceptions
Chapter 6: Swing and Event Handling (7 Hours)
Topics Covered: 6.1 What is GUI?, AWT (Abstract Window Toolkit), Comparison between AWT and Swing. 6.2 The MVC Architecture and Swing 6.3 The JComponent class, Components – JButton, JLabel, JText, JTextArea, JCheckBox and JRadioButton, JList, JComboBox, JMenu and JPopupMenu Class, JMenuItem and JCheckBoxMenuItem, JRadioButtonMenuItem ,JScrollBar, Dialogs (Message, confirmation, input), JFileChooser, JColorChoose6.4r 6.4 Layout Manager and Layouts
6.5 Event Handling: Event sources, Listeners, Mouse and Keyboard Event Handling, Adapters

Teaching Methodology:
Lectures, PPT presentation, you-tube link sharing and discussion on related topic,
Use of Google classroom.
Assessment & Evaluation:
Assignment, viva and presentation
Bibliography:
1.E. Balagurusamy, Programming with Java, A primer, Fourth edition, Tata McGraw Hill Education Private Limited, 2009
2. Herbert Schildt, “Complete reference Java (5th edition)”, Tata McGraw Hill Edition, 2005
Webliography:
1. https://www.tutorialspoint.com
2. http://www.javatpoint.com
3. https://www.geeksforgeeks.com

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-302-MJ-TH	Web Technology-I	2	TH	CIE:20 SIE:30
Course Hours: 30					
Course Objectives:					
1. Introduce students to the fundamentals of web technologies including HTML, CSS, HTTP, and PHP. 2. Develop an understanding of client–server architecture, web servers, and web browsers. 3. Enable students to design and develop static and dynamic web pages using HTML5, CSS, and PHP. 4. Familiarize students with PHP programming concepts such as functions, strings, arrays, and file handling. 5. Provide practical knowledge of database connectivity and interaction using PHP. 6. Teach students the concepts of email communication and handling email services using PHP. 7. Build a foundation for developing web-based applications following standard programming practices.					
Course Outcomes: (COs)					
CO1: Describe the basic structure of HTML, CSS, HTTP, and PHP and identify common HTML tags, CSS concepts, and PHP syntax. CO2: Construct web pages using HTML forms, tables, and semantic elements and implement CSS for styling and layout, including navigation bars and box model. CO3: Define and invoke PHP functions, handle default and variable parameters and manipulate strings, perform comparisons, and use regular expressions for pattern matching. CO4: Distinguish between indexed, associative, and multidimensional arrays and traverse, sort, extract data from arrays, and convert between arrays and variables. CO5: Read, write, and manipulate files and directories using PHP and access relational databases using SQL and PHP, and apply advanced database techniques. CO6: Explain the structure of an email message and Internet mail protocols and send emails and validate email IDs using PHP scripts effectively.					
Syllabus:					
Chapter 1 : Introduction to HTML, HTTP and PHP (9 Hours)					
Topics Covered:					

1.1 Overview of HTML and Basic Tags 1.1.1 Creating Forms ,Tables 1.1.2 HTML5 Semantics 1.2 CSS basic concept 1.2.1 Three ways to use CSS 1.2.2 Box Model, Navigation Bar. 1.3 Introduction to Web server and Web browser 1.4 HTTP basics. 1.5 PHP Basics 1.5.1 Use of PHP 1.5.2 Lexical structure 1.5.3 Language basics .
Chapter 2 : Function and String (7 hours)
Topics Covered: 2.1 Defining and calling a function 2.1.1 Default parameters Variable parameters, Missing parameters 2.1.2 Variable function, Anonymous function 2.2 Types of strings in PHP 2.2.1 Printing functions 2.2.2 Encoding and escaping 2.2.3 Comparing strings 2.2.4 Manipulating and searching strings 2.2.5 Regular expressions
Chapter 3 : Arrays (6 Hours)
Topics Covered: 3.1 Indexed Vs Associative arrays 3.2 Identifying elements of an array 3.3 Storing data in arrays 3.4 Multidimensional arrays 3.5 Extracting multiple values from arrays 3.6 Converting between arrays and variables 3.7 Traversing arrays 3.8 Sorting Action on entire array
Chapter 4 : Files and database handling (6 Hours)
Topics Covered: 4.1 Working with files and directories 4.1.1 Opening and Closing 4.1.2 Getting information about file 4.1.3 Read/write to file 4.1.4 Splitting name and path from file 4.1.5 Rename and delete files 4.1.6 Reading and writing characters in file 4.1.7 Reading entire file 4.1.8 Random access to file data 4.1.9 Getting information on file 4.1.10 Ownership and permissions 4.2 Using PHP to access a database Relational databases and SQL 4.2.1 PEAR DB basics 4.2.3 Advanced database techniques

Chapter 5 : Handling email with PHP (2 Hours)
Topics Covered: 5.1 Email background 5.2 Internet mail protocol 5.3 Structure of an email message 5.4 Sending email and validation of Email-id with PHP
Teaching Methodology:
1. Lecture Method 2. PPT presentation 3. Demonstration-Based Learning 4. Problem-Solving Approach 5. Activity-Based Learning
Assessment & Evaluation:
As mentioned above in the Evaluation and Assessment section.
Bibliography:
1. HTML & CSS: The Complete Reference, Fifth Edition Author: Thomas A. Powell First published: 01 Jan 2010. 2. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication 3. Beginning PHP 5 , Wrox publication 4. PHP web sevice, Wrox publication 5. Mastering PHP , BPB Publication 6. PHP cookbook, O'Reilly publication 7. PHP for Beginners, SPD publication 8. Programming the World Wide Web , Robert W Sebesta(3rd Edition) 9. HTML 5 Black Book : Covers Css3, Javascript, XML, XHTML, Ajax, PHP And JQuery by Kogent Learning Solutions Inc, Published November 2011 by Dreamtech Press 10. Spurlock Jake, Bootstrap: Responsive Web development. O'Reilly Media, Inc
Webliography:
1. www.php.net.in 2. www.W3schools.com 3. www.wrox.com 4. https://coreui.io/docs/layout/grid/#grid-options 5. https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-303-MJ-TH	Operating Systems	2	TH	CIE :20 SEE :30
Course Hours: - 30					
Course Objectives:					
1. To understand the concept of operation system and its principle					
2. To study the various functions and services provided by operating system					
3. To understand the notion of process and threads					
4. To understand the issue of Deadlocks in Process management.					
5. To understand the concept of File system management & disk scheduling					
Course Outcomes: (COs)					
1. Processes and Thread Scheduling by operating system					
2. Synchronization in process and threads by operating system					
3. Memory management by operating system using with the help of various schemes					
4. Management of deadlocks and File System by operating system					
5. Scheduling storage or disk for processes					
Syllabus:					
Chapter 1 : Introduction to Operating Systems (04 Hours)					
Topics Covered:					
1.1. Operating Systems Overview- system Overview and Functions of operating systems					
1.2. Types of Operating Systems					
1.3. What does an OS do?					
1.4. Operating system Operations					
1.5. Operating system structure					
1.6. Protection and security					
1.7. Computing Environments- Traditional, mobile, distributed, Client/server, peer-to-peer computing					
1.8. Open-source operating System					
1.9. Booting					
1.10. Operating System Services, User Mode vs Kernel Mode					
1.11. System calls: Types of System calls and their working					
Chapter 2 : Process Management (07 Hours)					
Topics Covered:					
2.1. Process Concept, PCB, States					
2.2. Threads, Multithreading, Types of Threads					
2.3. Process Scheduling (CPU Scheduling)					
2.3.1 Preemptive & non-preemptive					
2.3.2 FCFS, SJF, Priority, Round Robin					
2.4. Inter-process Communication (IPC)					
2.4.1. Shared Memory, Message Passing, Pipes, Sockets					
2.5. Context Switching					
Chapter 3 : Synchronization & Deadlocks (05 Hours)					

Topics Covered: 3.1. Critical Section Problem 3.2. Synchronization Mechanisms 3.2.1. Semaphores, Mutexes, Monitors 3.3. Classical Problems 3.3.1. Producer–Consumer 3.3.2. Dining Philosophers 3.3.3. Readers–Writers 3.4. Deadlock 3.4.1. Necessary Conditions 3.4.2. Prevention, Avoidance (Banker's Algorithm) 3.4.3. Detection & Recovery
Chapter 4 : Memory Management (06 Hours)
Topics Covered: 4.1. Address binding, Logical vs Physical Address Space 4.2. Dynamic loading, Dynamic linking and shared libraries 4.3. Contiguous Memory Allocation 4.4. Swapping 4.5. Paging (Basic Method, Hardware support, Protection, Shared Pages) 4.6. Segmentation - (Basic concept, Hardware) 4.7. Virtual Memory Concepts - (Background, Demand paging, Performance of demand paging,) 4.8. Page Replacement Algorithms 4.8.1. FIFO, LRU, MRU, Optimal
Chapter 5 : File System Management (04 Hours)
Topics Covered: 5.1. File Concepts, File Attributes 5.2. Access Methods (Sequential, Direct, Indexed) 5.3. Directory Structures 5.4. Allocation Methods 5.4.1. Contiguous, Linked, Indexed 5.5. Free Space Management - Bit vector, Linked list, Grouping, Counting, Space maps
Chapter 6 : Storage & Disk Management (04 Hours)
Topics Covered: 6.1. Disk Structure, Formatting 6.2. Disk Scheduling Algorithms 6.2.1. FCFS, SSTF, SCAN, C-SCAN, LOOK
Teaching Methodology:
Lecture, Board, PPT, Problem Solving
Assessment & Evaluation:
Internal : 20 Internal Exam :10 + Assignment / OBT / Viva :10 External :30

Bibliography:

1. Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia
2. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India.
3. Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001
4. The 'C' Odyssey, UNIX-the open boundless C, Meeta Gandhi, Tilak Shetty, Rajiv Shah, BPB publication
5. Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001 (Text Book)
6. Distributed Operating Systems Concepts and Design, Pradeep K. Sinha, PHI
7. Prasant Kumar Pattnaik, Rajib Mall, "Fundamentals of Mobile Computing", PHI Learning Pvt.Ltd, New Delhi – 2012.
8. Tannenbum, Herbert Bos, "Modern Operating systems", Pearson Publication, 4th Edition
9. Tannenbum, Maarten van Steen, "Distributed systems", 3rd Edition
10. Source wikipedia, Mobile operating systems, General books, LLC, 2010

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-304-MJ- TH	Theory of Computer Science	2	TH	CIE:20 SEE:30
Course Hours: 30					
Course Objectives:					
1. To understand the Finite Automata, Pushdown Automata and Turing Machine.					
2.To understand the Regular Language, Context Free Language, Context Sensitive Language and Unrestricted Language.					
3.To understand the relation between Automaton and Language					
Course Outcomes: (COs)					
1. On completion of the course, student will be able to–					
2. Understand the use of automata during language design.					
3. Relate various automata and Languages.					
Syllabus:					
Chapter 1 : Finite Automaton (10 Hours)					
Topics Covered:					
1.1 Introduction: Symbol, Alphabet, String, Prefix & Suffix of Strings, Formal Language, Operations on Languages.					
1.2 Deterministic finite Automaton – Definition, DFA as language recognizer, DFA as pattern recognizer. Nondeterministic finite automaton – Definition and Examples.					
1.3 NFA to DFA					
1.4 NFA with ϵ - transitions Definition					
1.5 NFA with ϵ -Transitions to DFA					
1.6 Finite automaton with output – Introduction to Mealy and Moore machine on Definition					
1.7 Definition and Examples. Minimization of DFA, Algorithm & Problem using Table Method. (Myhill Nerode Method)					
Chapter 2 : Regular Expressions and Languages (4 Hours)					
Topics Covered:					
2.1 Regular Expressions (RE): Definition & Example Regular Expressions Identities.					
2.2 Regular Language-Definition and Examples.					
2.3 Conversion of RE to FA-Examples. Pumping lemma for regular languages and applications. Closure Properties of regular Languages					
Chapter 3 : Context-Free Grammars and Languages (9 Hours)					

Topics Covered: 3.1 Grammar - Definition and Examples. Derivation-Reduction - Definition and Examples. 3.2 Chomsky Hierarchy. CFG: Definition & Examples. LMD, RMD, Parse Tree Ambiguous Grammar: Concept & Examples. 3.3 Simplification of CFG: 3.3.1 Removing Useless Symbols 3.3.2 Unit Production 3.3.3 ϵ-production and Null able Symbol. 3.4 Normal Forms: Chomsky Normal Form (CNF) and Greibach Normal Form (GNF) only Definition 3.5 Regular Grammar: Definition. Left linear and Right Linear Grammar-Definition and Example. 3.6 Equivalence of FA & Regular Grammar Construction of regular grammar equivalent to a given DFA.
Chapter 4 : Push Down Automata (3 Hours)
Topics Covered: 4.1 Definition of PDA and examples. 4.2 Construction of PDA using empty stack and final State method: Examples using stack method. 4.3 Definition DPDA & NPDA
Chapter 5 : Turing Machine (4 Hours)
Topics Covered: 5.1 The Turing Machine Model, Definition and Design of TM Problems on language recognizers, Language accepted by TM. 5.2 Types of Turing Machines (Multitrack TM, Two-way TM, Multitape TM, Non- deterministic TM) 5.3 Introduction to LBA (Basic Model) & CSG. (Without Problems)
Teaching Methodology: Lecture, Problem Solving, Demonstration, Class Test, Revision
Assessment & Evaluation:
Internal Evaluation :- 20 Marks
External Evaluation:- 30 Marks
Bibliography:
1. Introduction to Automata Theory, Languages and Computation, John E. Hopcraft, Rajeev Motwani, Jeffrey D. Ullman, Third Edition, Pearson Education Publication, 2008 2. Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House, 1995 3. Theory of Computer Science Automata, Languages and Computation, K.L.P. Mishra, N. Chandrasekaran, Publication- Prentice Hall of India, 2008 4. Introduction to Computer Theory Daniel I. A. Cohen – 2nd edition – John Wiley & Sons, 1996 5. Introduction to Languages and The Theory of Computation John C. Martin The McGraw- Hill, Fourth Edition, 2011

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-305-MJ-PR	Lab Course based on CS-301-MJ-TH	2	PR	CIE-20 SEE-30
Course Hours: 60					
Course Objectives:					
1 List the basics of object-oriented programming.					
2. To learn implementation of object-oriented concepts					
3. To introduce the principles of inheritance and polymorphism and demonstrate how they relate to the design of abstract classes					
4. Implement file handling and exception Handling.					
5. Design GUI based applications using swing.					
Course Outcomes: (COs)					
1. Identify classes, objects, class members and relationships for a given problem.					
2. Implement different types of object-oriented concepts.					
3. Handle File and directories.					
4. Handle abnormal termination of a program using exception handling					
5. Use Java APIs for GUI development					
Assignment 1: Fundamentals of Java Programming					
Topics Covered: Introduction to the java environment Use of java tools like java, javac, jdb and javadoc Defining simple classes and creating objects.					
Assignment 2 : Classes and packages					
Topics Covered: Defining a class. Creating an array of objects. Creating a package.					
Assignment 3: Inheritance					
Topics Covered: To implement inheritance in java. To define abstract classes. To define and use interfaces and Functional Interface					
Assignment 4: String, Streams and File Handling					
Topics Covered: Creation of files String methods implementation demonstration of I-O operations					

Assignment 5: Exception Handling
Topics Covered: Demonstrate Exception Handling Mechanism in Java. Use of try, catch, throw, throws, finally blocks Defining User defined Exception classes.
Assignment 6: Swing and Event Handling
Topics Covered: To demonstrate GUI creation using Swing Package and Layout managers. To understand Event handling mechanism in Java. Using Event classes, Event Listeners and Adapter
Teaching Methodology:
Lectures, PPT presentation, You-tube link sharing and discussion on related topic,
Use of Google classroom.
Assessment & Evaluation:
Assignment, Viva and Presentation
Bibliography:
1.E. Balagurusamy, Programming with Java, A primer, Fourth edition, Tata McGraw Hill Education Private Limited, 2009
2. Herbert Schildt, "Complete reference Java (5th edition)", Tata McGraw Hill Edition, 2005
Webliography:
1. https://www.tutorialspoint.com
2. http://www.javatpoint.com
3. https://www.geeksforgeeks.com

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-306-MJ-PR	Lab Course based on CS-302-MJ-TH	2	PR	CIE:20 SEE:30
Course Hours:60					
Prerequisites:					
Fundamentals of Computer Systems - Understanding of computer hardware and software concepts - Familiarity with operating systems and file management Basic Programming Concepts - Knowledge of variables, data types, operators, and control structures - Basic understanding of functions and modular programming Introduction to Internet and Web Concepts - Awareness of how the internet works - Basic understanding of websites and web browsers Database Fundamentals (Basic Level) - Understanding of data, records, and tables - Basic idea of databases and SQL concepts Logical Thinking and Problem-Solving Skills - Ability to analyze problems and implement solutions using programming logic					
Course Objectives:					
To introduce web development technologies - To provide hands-on experience in creating web pages using HTML and CSS - To understand page structure, styling, and layout techniques To develop responsive web applications - To use Bootstrap framework for designing responsive and mobile-friendly websites - To apply modern UI components and layout systems To strengthen programming skills - To understand the use of functions, strings, and arrays in programming - To develop efficient, modular, and reusable code To introduce file handling concepts - To perform file input and output operations - To understand data storage using files To provide database management experience - To design and implement databases using PostgreSQL - To perform CRUD operations (Create, Read, Update, Delete) - To integrate programming applications with database systems To enhance problem-solving and practical skills - To apply theoretical concepts in real-time programming scenarios - To develop debugging, testing, and documentation skills					

Course Outcomes: (COs)
CO1: Recall and explain the fundamentals of HTML, HTML5, CSS, and Bootstrap and identify different HTML tags, CSS properties, and Bootstrap components for web design. CO2: Apply CSS and Bootstrap classes to create responsive and visually appealing web pages and demonstrate the use of Bootstrap grid system, utilities, and components in web design projects. CO3: Define and call functions in PHP and other programming languages for practical problems and manipulate strings using built-in functions and analyse outputs for correctness. CO4: Understand and differentiate between types of arrays (1D, 2D) and perform array operations such as insertion, deletion, searching, and sorting and evaluate array- based solutions to solve programming problems effectively. CO5: Perform file handling operations, including reading from and writing to files, and integrate file data with database operations and design and implement database tables, queries, and CRUD operations using PostgreSQL. CO6: Integrate web development concepts, PHP programming, file handling, and databases to create functional, interactive, and responsive applications and test, debug, and optimize programs and applications for real-world scenarios.
Assignment 1: HTML, HTML5 & CSS
Topics Covered: • HTML document structure and basic tags • Text formatting, lists, links, images, and tables • HTML5 semantic elements and forms • Introduction to CSS and types of CSS (inline, internal, external) • CSS selectors and basic styling (text, background, borders, layout) • Basic responsive design using CSS
Assignment 2: Bootstrap
Topics Covered: • Introduction to Bootstrap framework • Linking Bootstrap using CDN • Bootstrap grid system (container, row, columns) • Responsive web design using Bootstrap classes • Bootstrap components: buttons, navbar, cards, alerts, forms • Utility classes for spacing and alignment • Creating responsive web pages using Bootstrap
Assignment 3: Functions and Strings
Topics Covered: • Concept and types of functions • Function declaration, definition, parameters, and return values • Variable scope • String declaration and basic operations (concatenation, length calculation, comparison) • Built-in string functions and string manipulation
Assignment 4: Arrays
Topics Covered: • Introduction to arrays • Indexed arrays and associative arrays • Array declaration and initialization • Traversing arrays using loops • Array operations: insertion, deletion, searching, and sorting • Programs based on array manipulation using indexed and associative arrays

Assignment 5: Files and DATABASE (PostgreSQL)	
Topics Covered:	
• File handling: open, close, read, write, and append operations • Introduction to PostgreSQL database • Database and table creation • Basic SQL commands: INSERT, SELECT, UPDATE, DELETE • Connecting programs with PostgreSQL database	
Teaching Methodology:	
1. Conceptual Explanation 2. Demonstration Method 3. Hands-on Practice 4. Step-by-Step Guidance 5. Assignment-Based Learning 6. Discussion and Doubt Solving 7. Continuous Evaluation	
Assessment & Evaluation:	
1. As mentioned above in the Evaluation and Assessment section.	
Bibliography:	
1. Duckett, J. HTML and CSS: Design and Build Websites, John Wiley & Sons, 2011. 2. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development, John Wiley & Sons, 2014. 3. W3C. HTML5 Specification, World Wide Web Consortium, Latest Edition. 4. Silberschatz, A., Korth, H. F., & Sudarshan, S. Database System Concepts, 6th Edition, McGraw-Hill Education, 2019. 5. Elmasri, R., & Navathe, S. B. Fundamentals of Database Systems, 7th Edition, Pearson Education, 2016. 6. Deitel, P., & Deitel, H. Internet and World Wide Web: How to Program, 5th Edition, Pearson Education, 2012. 7. Date, C. J. An Introduction to Database Systems, 8th Edition, Pearson Education, 2003.	
Webliography:	
1. W3Schools https://www.w3schools.com Reference for HTML, CSS, Bootstrap, JavaScript, and SQL basics. 2. MDN Web Docs (Mozilla) https://developer.mozilla.org Authoritative documentation for HTML, CSS, and web standards. 3. Bootstrap Official Documentation https://getbootstrap.com Guidelines and examples for responsive web design using Bootstrap. 4. PostgreSQL Official Documentation https://www.postgresql.org/docs Complete reference for PostgreSQL installation, SQL queries, and database management.	

5. **GeeksforGeeks**
<https://www.geeksforgeeks.org> Programming concepts, functions, strings, arrays, and file handling.
6. **Tutorials Point**
<https://www.tutorialspoint.com>
Step-by-step tutorials on web technologies and databases.
7. **W3C Official Website**
<https://www.w3.org>
Web standards, specifications, and best practices

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-310-MJ-TH	Data Science	2	TH	CIE :20 SEE :30
Course Hours: - 30 Hours					
Prerequisites:					
• Understanding of programming fundamentals (preferably in Python / C / Java). • Familiarity with data structures such as arrays, lists, and basic file handling. • Basic understanding of mathematics, statistics - including: Descriptive statistics (mean, median, mode), Probability fundamentals, Linear algebra basics (vectors, matrices) • Familiarity with spreadsheets or data handling tools is desirable. • Knowledge of Databases					
Course Objectives:					
1. To Provide students with knowledge and skills for data-intensive problem solving and scientific discovery					
2. To Be prepared with a varied range of expertise in different aspects of data science such as data collection, visualization, processing and modeling of large data sets.					
3. To acquire good understanding of both the theory and application of applied statistics and					
4. To apply computer science-based existing data science models to analyze huge data sets					
5. To Be better trained professionals to cater the growing demand for data scientists in industry.					
Course Outcomes: (COs)					
1. Perform Exploratory Data Analysis					
2. Obtain, clean/process, and transform data					
3. Detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization					
4. Demonstrate proficiency with statistical analysis of data.					
5. Present results using data visualization techniques.					
6. Prepare data for use with a variety of statistical methods and models and recognize how the quality of the data and the means of data collection may affect conclusions.					
Chapter 1 : Introduction to Data Science (6 Hours)					
Topics Covered: 1.1. Introduction to data science, The 3 V's: Volume, Velocity, Variety 1.2. Why learn Data Science? 1.3. Applications of Data Science 1.4. The Data Science Lifecycle 1.5. Data Scientist's Toolbox 1.6. Types of Data - Structured, semi-structured, Unstructured Data, Problems with unstructured data 1.7. Data sources - Open Data, Social Media Data, Multimodal Data, standard datasets 1.8. Data Formats - Integers, Floats, Text Data, Text Files, Dense Numerical Arrays, Compressed or Archived Data, CSV Files, JSON Files, XML Files, HTML Files , Tar Files, GZip Files, Zip Files, Image Files: Rasterized, Vectorized, and/or Compressed					

Chapter 2 : Statistical Data Analysis (08 Hours)
Topics Covered: 2.1 Role of statistics in data science 2.2.Descriptive statistics Measuring the Frequency Measuring the Central Tendency: Mean, Median, and Mode Measuring the Dispersion: Range, Standard deviation, Variance, Interquartile Range 2.3.Inferential statistics Hypothesis testing, Multiple hypothesis testing, Parameter Estimation methods, 2.4 Measuring Data Similarity and Dissimilarity Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes, Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Euclidean, Manhattan, and Minkowski distances, Proximity Measures for Ordinal Attributes 2.5.Concept of Outlier, types of outliers, outlier detection methods
Chapter 3 : Data Preprocessing (8 Hours)
Topics Covered: 3.1 Data Objects and Attribute Types: What Is an Attribute?, Nominal, Binary, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes 3.2 Data Quality: Why Preprocess the Data? 3.3.Data munging/wrangling operations Cleaning Data - Missing Values, Noisy Data (Duplicate Entries, Multiple Entries for a Single Entity, Missing Entries, NULLs, Huge Outliers, Out-of- Date Data, Artificial Entries, Irregular Spacings, Formatting Issues - Irregular between Different Tables/Columns, Extra Whitespace, Irregular Capitalization, Inconsistent Delimiters, Irregular NULL Format, Invalid Characters 3.4 Data Transformation – Rescaling, Normalizing, Binarizing, Standardizing, Label and One Hot Encoding, Data reduction, Data discretization
Chapter 4 : Data Visualization (08 Hours)
Objectives: ● Be better trained professionals to cater the growing demand for data scientists in industry
Topics Covered: 4.1 Introduction to Exploratory Data Analysis 4.2 Data visualization and visual encoding 4.3 Data visualization libraries 4.4 Basic data visualization tools: Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts, 4.5 Specialized data visualization tools: Boxplots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots 4.6 Advanced data visualization tools- Wordclouds, Visualization of geospatial data, Data Visualization types
Teaching Methodology:
Lecture, Problem Solving, Case Study
Assessment & Evaluation:
Internal: 20
Internal Exam :10 + Assignment / OBT / Viva :10
External :30

Bibliography:
1) Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020.
2) The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017
3) Data Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline Kamber,
4) A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press
Webliography:
https://www.w3schools.com/datascience/ds_linear_functions_graphic.asp https://github.com/Adwait197/Data-Science-Practical https://www.scaler.com/topics/data-mining-tutorial/measures-of-similarity-and-dissimilarity

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-311-MJ-PR	Lab Course based on CS-310-MJ-TH	2	PR	CIE :20 SEE :30
Course Hours: - 60 Hours					
Course Objectives:					
1. To apply statistical techniques on data sets					
2. To apply data preprocessing techniques on data sets					
3. To apply data visualization techniques on data sets					
Course Outcomes: (COs)					
1. Prepare data for use with a variety of statistical methods					
2. Recognize how the quality of the data may affect conclusions					
3. Perform exploratory data analysis					
Assignment 1 : The Data Science environment					
Topics Covered:					
Getting introduced to Python IDLE, command line, online tools like Google Colaboratory, and essential packages like NumPy, SciPy, pandas, scikit-learn, matplotlib, Jupyter, beautifulsoup, etc.					
Assignment 2 : Loading the dataset					
Topics Covered:					
Select a dataset from a list of publicly available datasets at UCI Machine Learning Repository and load it using Pandas. (Import different dataformat files like .CSV,.htm,.json etc. Briefly describe what the dataset is about and size of the dataset (e.g. number of tables, number of instances and attributes, etc.)					
Assignment 3 : Basic statistical operations					
Topics Covered:					
Apply basic statistical operations on a dataset. For example - compute the mean, median, mode, range, quartiles, and variance for one or more attributes.					
Assignment 4 : Data Processing					
Topics Covered:					
Apply data preprocessing techniques that are likely required for the dataset.					
1)Partition them into appropriate number of bins by equal-frequency as well as equal-width partitioning.					
2)Use smoothing by bin means to smooth the data based on the above partitioning, 3)Normalize the attribute based on min-max normalization and z-score normalization. Comment on which method you would prefer to use for partitioning, smoothing, and					
normalization for the given attribute.					

Assignment 5 : Data Visualization with matplotlib
Topics Covered:
View the data using various 2-D, 3-D plots and charts, setting styles, saving the figures, customizing the legends, multiple subplots,
Teaching Methodology:
Demo, Explanation
Assessment & Evaluation:
Internal : 20
External :30
Bibliography:
1. Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam
2. Sharma, BPB Publications, 2020.
3. The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017
4. Ta Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline Kamber,
5. A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press
Webliography:
https://www.w3schools.com/datascience/ds_linear_functions_graphic.asp https://github.com/Adwait197/Data-Science-Practical https://www.scaler.com/topics/data-mining-tutorial/measures-of-similarity-and- dissimilarity

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-312-MJ-TH	Database Technologies	2	TH	CIE- 20 SEE- 30
Course Hours: 30					
Prerequisites:					
Basic knowledge of database.					
Course Objectives:					
1. Understand the limitations of relational databases and the need for NoSQL					
2. Learn document-oriented database concepts using MongoDB					
3. Design efficient and scalable data models					
4. Perform advanced data retrieval and aggregation					
5. Encourage practical exposure through hands-on exercises using MongoDB.					
Course Outcomes: (COs)					
On successful completion of the course, students will be able to:					
1. Explain NoSQL concepts and MongoDB architecture.					
2. Explain the fundamentals of NoSQL databases and justify the need for MongoDB over traditional relational databases.					
3. Design document-based data models using MongoDB					
4. Perform CRUD operations using MongoDB queries.					
5. Apply indexing and aggregation for performance optimization					
Chapter 1 : Introduction to NoSQL and MongoDB (4 Hours)					
Topics Covered:					
1. Limitations of Relational Databases					
2. Introduction to NoSQL Databases					
3. Types of NoSQL Databases					
4. Advantages and Use Cases of NoSQL					
5. Introduction to MongoDB and its Applications					
Chapter 2 : MongoDB Architecture and Data Model (5 Hours)					
Topics Covered:					
1. MongoDB Overview - Documents, Collections, and Databases					
2. BSON Data Format					
3. MongoDB Architecture Components					
4. MongoDB vs RDBMS					
Chapter 3 : Installation and MongoDB Tools (2 Hours)					
Topics Covered:					
1. Installing MongoDB Community Edition					
2. MongoDB Shell (mongosh)					
3. MongoDB Compass (GUI Tool)					
4. Creating and Connecting to Databases					
Chapter 4 : MongoDB Data Modelling (5 Hours)					

Topics Covered:
1. Document Structure 2. Schema Design Concepts 3. Embedded Documents 4. References 4.1 One-to-One Relationship One-to-Many Relationship
Chapter 5 : CRUD Operations in MongoDB (7 Hours)
Topics Covered:
1. Creating Databases and Collections 1.1 - Insert Operations (insertOne, insertMany) 1.2 - Read Operations (find, findOne) 2. Query Operators 2.1 Update Operations (updateOne, updateMany) Delete Operations (deleteOne, deleteMany)
Chapter 6: Querying and Indexing (4 Hours)
Topics Covered:
1. Comparison and Logical Operators 1.1 Projection, Sorting, and Limiting 2. Regular Expressions 3. Introduction to Indexes Creating and Dropping Indexes
Chapter 7: Aggregation Framework (3 Hours)
Topics Covered:
1. Introduction to Aggregation Aggregation Pipeline - \$match, \$group, \$project, \$sort, \$limit
Teaching Methodology:
Lecture, Demonstration, Discussion, Case Study
Assessment & Evaluation:
Class Test/Group Discussion/PPT/Assignment, Internal Exam
Bibliography:
1. MongoDB: The Definitive Guide by Kristina Chodorow 2. NoSQL Distilled by Martin Fowler 3. MongoDB Basics by David Hows, Peter Membrey and Eelco Plugge NoSQL with MongoDB in 24 Hours, Sams Teach Yourself
Webliography:
1. MongoDB Official Documentation 2. MongoDB Tutorial 3. MongoDB Introduction - GeeksforGeeks

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-313-MJ-PR	Lab Course based on CS-312-MJ-TH	2	PR	CIE-20 SEE-30
Course Hours: 30 Hours					
Prerequisites:					
Basic knowledge of database.					
Course Objectives:					
1. To understand NoSQL database concepts and MongoDB architecture					
2. To perform CRUD operations using MongoDB					
3. To apply query operators, indexing, and aggregation					
4. To design data models using embedded and referenced documents					
Course Outcomes: (COs)					
On successful completion of the course, students will be able to:					
1. Explain NoSQL concepts and MongoDB fundamentals					
2. Create and manage databases and collections					
3. Perform CRUD operations on documents					
4. Apply query operators and data retrieval techniques					
5. Use indexing and aggregation for performance and analysis					
Assignment 1: MongoDB basics					
Objectives:					
To study NoSQL databases and MongoDB features and to install MongoDB and MongoDB Compass.					
Topics Covered:					
- MongoDB Installation process - MongoDB Compass installation					
Assignment 2: MongoDB Create, Read operations and Operators					
Objectives:					
To study creation of MongoDB document with insert command and perform different comparison operations using MongoDB operators.					
Topics Covered:					
- Commands to create, insert and delete collection. - Relationships between documents - MongoDB datatypes, operators - Executing different queries on MongoDB database					
Assignment 3: Update and Delete operation in MongoDB					
Objectives:					
To study updating of MongoDB document and delete operation on MongoDB document.					

Topics Covered: • Methods to perform Update and delete operations on MongoDB database • Executing different queries on MongoDB database
Assignment 4: MongoDB Index and Aggregation
Objectives: • To apply query operators, indexing, and aggregation
Topics Covered: • Different commands to perform different operations on index • Aggregation commands • Executing different queries on MongoDB database
Teaching Methodology:
Demonstration method, Hands on Practical method, Problem Solving method, Assignment and practice method
Assessment & Evaluation:
Lab book Assignment
Bibliography:
1. MongoDB: The Definitive Guide by Kristina Chodorow 2. NoSQL Distilled by Martin Fowler 3. MongoDB Basics by David Hows, Peter Membrey and Eelco Plugge 4. NoSQL with MongoDB in 24 Hours, Sams Teach Yourself
Webliography:
1. MongoDB Official Documentation 2. MongoDB Tutorial 3. MongoDB Introduction - GeeksforGeeks

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-321-VSC-PR	Lab Course based on CS-303-MJ-TH	2	PR	CIE :20 SEE :30
Course Hours: - 60					
Course Objectives:					
1. To understand the concept of the operation system and its principle					
2. To study the various functions and services provided by operating system					
3. To understand the notion of process and threads					
4. To understand the issue of Deadlocks in Process management.					
5. To understand the concept of File system management & disk scheduling					
Course Outcomes: (COs)					
1. Processes and Thread Scheduling by operating system					
2. Synchronization in process and threads by operating system					
3. Memory management by operating system using with the help of various schemes					
4. Management of deadlocks and File System by operating system					
5. Scheduling storage or disk for processes					
Assignment 1 : Operations on Processes					
Objectives:					
- To Understand the Operating System Overview - To Understand the Process Structure - To Learn the System call functions					
Topics Covered:					
System calls: Types of System calls and their working					
Assignment 2 : CPU Scheduling					
Objectives:					
- To Understand the Process Concept, PCB - To Learn Process Scheduling Algorithms (CPU Scheduling)					
Topics Covered:					
- Process Scheduling - Preemptive & Non-preemptive - FCFS, SJF, Priority, Round Robin					
Assignment 3 : Synchronization & Deadlocks					
Objectives:					
- To Understand the Critical Section Problem and Synchronization Mechanisms - To Understand the Different Classical Problems - To Learn Process Scheduling					
Topics Covered:					
Banker's Algorithm					
Assignment 4 : Demand Paging					

Objectives: - To Understand the Logical, Physical Memory Concept - To Understand the Paging Concept - To Learn Page Replacement Algorithms
Topics Covered: - Page Replacement Algorithms - FIFO, LRU, MRU, Optimal
Assignment 5 : File System Management
Objectives: - To Understand the File System Concept - To Understand the Allocation Methods and Free Space Management
Topics Covered: - Allocation Methods - Contiguous, Linked, Indexed
Assignment 6 : Disk Scheduling
Objectives: - To Understand the Storage and Disk Structure - To Understand the Disk Management - To Learn Disk Scheduling Algorithms
Topics Covered: - Disk Scheduling Algorithms - FCFS, SSTF, SCAN, C-SCAN, LOOK
Teaching Methodology:
Demo, Explanation, PPT,
Assessment & Evaluation:
Internal : 20
External :30
Bibliography:
- Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia - Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India. - Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001 - The ‘C’ Odyssey, UNIX-the open boundless C, Meeta Gandhi,Tilak Shetty,Rajiv Shah, BPB publication - Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001 (Text Book) - Distributed Operating Systems Concepts and Design, Pradeep K. Sinha, PHI - Prasant Kumar Pattnaik, Rajib Mall, “Fundamentals of Mobile Computing”, PHI Learning Pvt.Ltd, New Delhi – 2012. - Tannenbum, Herbert Bos, “Modern Operating systems”, Pearson Publication, 4th Edition - Tannenbum, Maarten van Steen, “Distributed systems”, 3rd Edition - Source wikipedia, Mobile operating systems, General books, LLC,2010

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-331-FP-PR	Project	4	PR	CIE-20 SEE-30
Course Hours: 30					
Course Objectives:					
1. Apply computer science principles to build real-world systems.					
2. Demonstrate project planning, design, and development skills.					
3. Use modern technologies, frameworks, and tools effectively.					
4. Communicate technical work clearly in written and oral form.					
5. Work collaboratively and professionally in a team environment.					
6. Evaluate ethical, social, and security impacts of computing solutions.					
Course Outcomes: (COs)					
1. Analyse a real-world problem and define clear technical requirements.					
2. Design an effective system architecture and development plan.					
3. Implement the system using appropriate tools and technologies.					
4. Evaluate, test, and refine the solution based on performance and feedback.					
5. Apply project management practices and maintain documentation.					
6. Communicate project ideas and results through reports and presentations.					
Project Implementation Guidelines for Students:					
1. Students shall choose any topic for project work in consultation with project guide, Project In-charge.					
2. The students shall work on a Project in a group of with at least three students.					
3. Students are expected to work on the chosen project during the entire semester.					
4. Students shall undertake application oriented/web-based with database/Survey-based work.					
5. Students shall successfully implement the chosen work. Only a hypothetical / theoretical study shall not be accepted.					
6. Students shall choose any appropriate programming language/ platform, computational techniques and tools in consultation with the guide, In-charge.					
7. The project work shall be evaluated based on the novelty of the topic, scope of the work, relevance to computer science, adoption of emerging techniques/technologies and its real-world application etc.					
8. Students shall prepare a project report with prescribed format provided by Project Guide.					
9. The Project report should be prepared in a spiral bound form with adequate number of copies.					
10. The Project work and report shall be certified by the concerned Project guide and Head of the department.					

Recommended Documentation contents:

Sr. No.	Index Name	Page No.
1.	Acknowledgement	
2.	College Certificate	
3.	Introduction I. Existing System II. Limitations of Existing System	
4.	Problem Definition	
5.	Proposed System: I. Explanation II. Methodology used III. Scope of Project	
6.	Hardware and Software Requirement	
7.	Fact Finding Techniques	
8.	Feasibility Study I. Operational II. Technical III. Economical	
9.	Diagrams: I. E-R Diagram II. UML Diagrams	
10.	Data Dictionary	
11.	Database Designing	
12.	Screen Designing I. I/O Screen Designing II. Output Formats	
13.	Test Cases Design	
14.	Conclusions and Future Enhancements	
15.	Bibliography, References and Published work (Paper/Book Chapter/Copyright/Patent etc.)	

Teaching Methodology:**Assessment & Evaluation:**

Description	Evaluation	Marks
Continuous Evaluation, Progress Report	Internal	20
Presentation & Project Report	External	10
Demonstration of the Project		15
Viva		5
Total		50

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	MTC-341-MN-TH	Optimization Techniques	2	TH	SEE: 30 CIE: 20
Course Hours: 30					
Course Objectives:					
1. Provide students with a comprehensive understanding of linear programming and optimization models, along with the various techniques used to solve them.					
2. Learn both the theory and practical applications of linear programming, including graphical methods, simplex methods, duality, and the handling of assignment and transportation problems.					
3. Be equipped with the tools to apply linear programming techniques in real-world optimization problems across various industries.					
Course Outcomes: (COs)					
After successful completion of this course, students will be able to:					
CO1: Recall and explain the fundamental concepts, definitions, and assumptions of Linear Programming Problems, including slack, surplus, and artificial variables					
CO2: Formulate and solve Linear Programming Problems using graphical method, simplex method, and Big-M method.					
CO3: Analyze Linear Programming models, Transportation problem and understand their applications, advantages, and limitations in practical situations.					
CO4: Apply Assignment and Transportation models using Hungarian method, North-West rule, Matrix Minima, Vogel's Approximation, and MODI method.					
Syllabus:					
Unit-1: Linear Programming Problem I					
1.1 Introduction Definition and Examples Problem solving using Graphical method					
Unit-2: Linear Programming Problem II					
2.1 Formation of LPP 2.2 Standard form of LPP, Slack and surplus variables, Some important definitions, Assumptions in LPP, 2.3 Simplex method, Big- M-method 2.4 Duality in Linear Programming 2.5 Limitations of Linear programming, Applications of Linear programming, Advantages of Linear Programming Techniques					
Unit-3: Assignment Models					
3.1 Assignment Model -Introduction Hungarian method for Assignment problem					
Unit-4: Transportation Models					
4.1 Introduction, Tabular representation 4.2 Methods of IBFS (North-West rule, Matrix-minima, Vogel's Approximation), Algorithms The Optimality Test of Transportation Model (MODI method only)					

Reference Books:

1. Kanti Swarup, P.K. Gupta, Man Mohan, Operations Research Sultan Chand & Sons, New Delhi.
 2. V. K. Kapoor
Operations Research – Problems and Solutions Sultan Chand & Sons.
 3. J. K. Sharma, Operations Research: Theory and Applications, Macmillan India Ltd.
- Hamdy A. Taha Operations Research: An Introduction, Pearson Education.

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	ELC-341-MN-TH	Digital Communication and Networking	2	TH	SEE: 30 CIE: 20
Course Hours: 30					
Course Objectives:					
- To introduce to all aspects of digital communication system - To introduce various digital modulation schemes - To identify the need of data coding and error detection/correction mechanism. - To study bandwidth utilization techniques: multiplexing and Spectrum spreading - To study OSI and TCP/IP models of Networking.					
Course Outcomes: (COs)					
CO1- Define and explain terminologies of digital communication					
CO2- Understand the impact and limitations of various digital modulation techniques					
CO3- Analyze different modulation multiplexing techniques.					
CO4- Build the computer network by Choosing appropriate devices and advanced techniques.					
Syllabus:					
Unit-1: Fundamentals of Digital Communication					07 Hours
Elements of Communication System: Block diagram of communication system (Transmitter, Receiver, Medium, Message) explain with Noise and Electromagnetic spectrum. Modes of Communication: Simplex, Half duplex, Full duplex, Baseband, Broadband, Serial, Parallel, Asynchronous and Synchronous, Information Theory: Data rate, Baud rate, Channel Capacity, Nyquist theorem, Signal to Noise ratio, Shannon theorem, Channel Bandwidth. Error Detection and Correction: Hamming Code, Cyclic Redundancy Check (CRC)					
Unit-2: Modulation and Demodulation Techniques					06 Hours
Introduction to modulation and demodulation: Concept and need of modulation and demodulation Digital Modulation techniques: Pulse Code Modulation (PCM), FSK, BPSK, QPSK, PWM (Block diagram of transmitter and receiver with its working principle, Advantages and disadvantages)					
Unit-3: Multiplexing and Multiple Access Technique					09 Hours
Multiplexing techniques: Frequency Division Multiplexing, Wavelength Division Multiplexing, Time Division Multiplexing Multiple Access techniques: Need, Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA) Spread Spectrum techniques: Pseudo-Noise (PN) sequences: Definition, generation and maximum length sequence, Types: Frequency hopping Spread Spectrum, Direct Sequence Spread					

Spectrum.
Unit-4: Computer Networking 08 Hours
• Network devices: Repeater, Switch, Networking cables, Router, Bridge, Hub, Brouter, Gateway. • Wired LANs: - Ethernet: Ethernet protocol, standard Ethernet, 100 MBPS Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet • Types of Networks: LAN, MAN, WAN, Wireless LAN, Internet, • Network topology: point to point, Star, Ring, Bus, Mesh, Tree, Hybrid • Computer Network model: OSI and TCP/IP. • Switching: Circuit-switched network, Packet switched network. (explain with TDM)
Reference Books:
1. Principles of Communication Systems, Taub & Schilling, (2nd Edition), Tata McGraw Hill Publication 2. Communication Electronics: Principles and Applications, Frenzel, Tata Mc Graw Hill publication 3. Data Communication and Networking, Forouzan, Mc Graw Hill publication, 5th edition 4. Computer Networks, Tanenbaum, pHI publication 5. Data Communication and Networks: Godbole Achyut, McGraw-Hill Higher Education ISBN-13 978-0-07- 296775-3 6. Computer Networks, A.S. Tanenbaum, PRENTICE HALL ISBN-10: 0-13-212695-8, ISBN13:978-0-13-212695-3 7. Digital Communications, Simon Haykin, John Wiley and Sons, ISBN-9788126508242. William Stallings, “Data and Computer Communication”
Examination Scheme:
Internal Evaluation: 20 Marks (Internal Examination, Seminars, Presentation Assignments)
External Evaluation: 30 Marks (Final Theory Examination)
Reference Video Links:
For Fundamental and Advanced Concepts in the Subject
<i>The following video resources are recommended to enhance both basic understanding and in-depth knowledge of the subject. These materials serve as supplementary learning tools for theoretical and practical insights.</i>
1. http://digimat.in/nptel/courses/video/117105144/L01.html 2. https://www.youtube.com/watch?v=7zRkUa1eSOA 3. http://kcl.digimat.in/nptel/courses/video/117104099/L01.html 4. http://kcl.digimat.in/nptel/courses/video/117104117/L01.html 5. http://kcl.digimat.in/nptel/courses/video/117104121/L01.html 6. http://kcl.digimat.in/nptel/courses/video/117105131/L01.html 7. http://kcl.digimat.in/nptel/courses/video/117105136/L01.html 8. https://www.netacad.com/courses/introduction-iot?courseLang=en-US https://www.youtube.com/watch?v=Ho_d7WXUYf0

SEM-VI

Syllabus

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-351-MJ-TH	Advanced Java	2	TH	CIE-20 SEE-30
Course Hours: 30 Hours					
Course Objectives:					
1. To learn database programming using Java					
2. To study web development concept using Servlet and JSP					
3. To develop applications using multithreading					
4. To implement various Collections.					
Course Outcomes: (COs)					
On completion of the course, student will be able to–					
1. To access open database through Java programs using Java Data Base Connectivity (JDBC) and develop the application					
2. Understand and create dynamic web pages, using Servlets and JSP.					
3. Work with basics of framework to develop secure web applications.					
4. Understand the use of collections in various applications.					
Syllabus:					
Chapter 1: Java Database Programming-JDBC (6 Hours)					
Topics Covered:					
1.1 The design of jdbc, jdbc configuration					
1.2 Types of drivers					
1.3 Executing sql statements, query execution					
1.4 Scrollable and updatable result sets					
1.5 Metadata – DatabaseMetadata, ResultSetMetadata					
1.6 Transactions – commit(), rollback(), SavePoint (Database : PostgreSQL)					
Chapter 2: Collections(4 Hours)					
Topics Covered:					
2.1 Introduction to the Collection framework					
2.2 List – ArrayList, LinkedList and Vector, Stack, Queue					
2.3 Set - HashSet, TreeSet, and LinkedHashSet					
2.4 Map – HashMap, LinkedHashMap, Hashtable and TreeMap					
2.5 Interfaces such as Comparator, Iterator, ListIterator, Enumeration					
Chapter 3: Multithreading(4 Hours)					
Topics Covered:					
3.1 What are threads?					
3.2 Life cycle of thread					
3.3 Running and starting thread using Thread class					
3.4 Thread priorities					
3.5 Running multiple threads					
3.6 The Runnable interface					
Synchronization and interthread communication					

Chapter 4: Servlet and JSP (10 Hours)
Topics Covered: 4.1 Introduction to Servlet and Hierarchy of Servlet 4.1.1 Life cycle of servlet 4.1.2 Handling get and post request (HTTP) 4.1.3 Handling data from HTML to servlet 4.1.4 Retrieving data from database to servlet 4.2 Session tracking – User Authorization, URL rewriting, Hidden form fields, Cookies and HttpSession 4.3 Introduction to JSP, 4.3.1 Life cycle of JSP 4.3.2 Implicit Objects 4.3.3 Scripting elements - Declarations, Expressions, Scriptlets, Comments 4.4 JSP Directives - Page Directive, include directive 4.5 Mixing Scriptlets and HTML 4.6 JSP Actions - jsp:forward , jsp:include, jsp:useBean, jsp:setProperty and jsp:getProperty
Chapter 5: Introduction to Spring framework(6 Hours)
Topics Covered: 5.1 Introduction of Spring framework 5.2 Spring Modules / Architecture 5.3 Spring Applications 5.4 Spring MVC 5.5 Spring MVC Forms, Validation
Teaching Methodology:
Lectures, Problem Solving, Discussion
Assessment & Evaluation:
Assignment and viva
Bibliography:
1. Java 2 programming black books, Steven Horlizer
2. Herbert Schildt, “Complete reference Java (5th edition)”, Tata McGraw Hill Edition, 2005
3. Getting started with Spring Framework: covers Spring 5 by J Sharma and Ashish Sarin
Webliography:
1. https://www.tutorialspoint.com
2. http://www.javatpoint.com
3. https://spring.io/projects/spring-framework
4. https://docs.oracle.com/javase/tutorial/

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-352-MJ-TH	Web Technology-II	2	TH	CIE:20 SEE:30
Course Hours: 30					
Prerequisites:					
• HTML5 • CSS • Core PHP • Bootstrap framework utility					
Course Objectives:					
1. Introduce students to advanced web techniques using PHP, XML, JavaScript, AJAX, and PHP frameworks. 2. Develop an understanding of server-side and client-side scripting and their interaction in web applications. 3. Enable students to process and validate web forms, manage sessions, cookies, and application state. 4. Provide knowledge of XML technologies and their integration with PHP. 5. Familiarize students with JavaScript and jQuery for dynamic user interface development. 6. Explain the working of AJAX-based web applications and asynchronous client-server communication. 7. Introduce students to MVC architecture using the CodeIgniter PHP framework. 8. Equip students with skills to develop interactive, database-driven web applications using modern web technologies.					
Course Outcomes: (COs)					
CO1: Recall and explain fundamental web techniques, including PHP variables, server information, form processing, response headers, and state management and understand XML structure, parsing methods, and how PHP interacts with XML documents. CO2: Apply JavaScript and jQuery concepts to implement interactive web pages, including control statements, functions, DOM events, and dynamic content manipulation and utilize AJAX to perform asynchronous requests, validate data, handle XML, and connect to databases using PHP. CO3: Develop web applications using CodeIgniter framework, implementing MVC architecture and handling dynamic data and analyse web application workflows to integrate PHP, JS, AJAX, and CodeIgniter components effectively. CO4: Evaluate and debug PHP programs, web forms, and AJAX requests for correctness, efficiency, and reliability and compare different methods of handling state, errors, and session management in web applications. CO5: Integrate PHP, XML, JavaScript/jQuery, AJAX, and CodeIgniter to design fully functional, responsive, and interactive web applications and combine client-side and server-side technologies to perform real-time updates and manage dynamic content. CO6: Critically assess web applications for performance, usability, and maintainability using modern web standards and PHP frameworks and design, implement, and test web solutions that leverage multiple technologies in a cohesive application.					
Syllabus:					

Chapter 1 : Introduction to Web Techniques (6 Hours)
Topics Covered: 1.1 Variables 1.2 Server information 1.3 Processing forms 1.4 Setting response headers 1.5 Maintaining state 1.6 PHP error handling
Chapter 2 : XML (6 Hours)
Topics Covered: 1.1 What is XML? 1.2 XML document Structure 1.3 PHP and XML 1.4 XML parser 1.5 The document object model 1.6 The simple XML extension 1.7 Changing a value with simple XML
Chapter 3 : Java Script and Jquery (8 Hours)
Topics Covered: 3.1 Overview of JavaScript 3.2 Object Orientation and JavaScript Basic Syntax(JS datatypes, JS variables) 3.3 Primitives, Operations and Expressions 3.4 Screen Output and keyboard input(Verification and Validation) 3.5 JS Control statements and JS Functions 3.6 JavaScript HTML DOM Events(onmouseup, onmousedown, onclick, onload, onmouseover, onmouseout). 3.7 JS Strings and JS String methods 3.8 JS popup boxes(alert, confirm, prompt). 3.9 Jquery library , Including jquery library in page 3.10 Jquery selector , DOM manipulation using jquery
Chapter 4 : AJAX (6 Hours)
Topics Covered: 4.1 Introduction of AJAX 4.2 AJAX web application model 4.3 AJAX –PHP framework 4.4 Performing AJAX validation 4.5 Handling XML data using php and AJAX 4.6 Connecting database using php and AJAX
Chapter 5 : PHP framework CodeIgniter (4 Hours)
Topics Covered: 5.1 CodeIgniter - Overview, Installing CodeIgniter 5.2 Application Architecture MVC Framework ,Basic concept of CodeIgniter, Libraries 5.3 Working with databases 5.4 Load external JS and CSS page & redirecting from controller , Adding JS and CSS , Page redirection. 5.5 Loading dynamic data on page & session management, cookies management

Teaching Methodology:

1.Lecture Method 2.Demonstration-Based Learning 3.Problem-Solving Approach 4. Activity-Based Learning
Assessment & Evaluation:
As mentioned above in the Evaluation and Assessment section.
Bibliography:
1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe O'Reilly publication 2. Beginning PHP 5, Wrox publication 3. AJAX Black Book Kogent solution 4. Mastering PHP BPB Publication 5. Professional Codeigniter By Thomas Myer ,Wrox Publication, 6. Codeihniter 2 CookBook By Rob Foster ,PACKT Publication , 7. JQuery CookBook, O'reilly Publication.
Webliography:
1. www.php.net.in 2. www.W3schools.com 3. https://www.tutorialspoint.com/codeigniter/index.htm 4. https://api.jquery.com/ 5. http://codeigniter.com/docs <i>link</i>

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
V	CS-353-MJ-TH	Design Framework - Node JS	2	TH	CIE:20 SEE:30
Course Hours: 30					
Course Objectives:					
1. Introduce students to the fundamentals of Node.js and its role in modern server-side web development.					
2. Explain the Node.js runtime environment , event-driven architecture, and non-blocking I/O model.					
3. Familiarize students with Node.js modules , core libraries, and package management using NPM .					
4. Enable students to develop web servers and handle HTTP requests and responses using Node.js.					
5. Provide hands-on knowledge of file system operations , event handling, and asynchronous programming					
6. Develop skills to connect Node.js applications with databases and perform basic CRUD operations					
Course Outcomes: (COs)					
1. Understand and explain the architecture, features, and advantages of Node.js.					
2. Install and configure the Node.js environment and use REPL for testing code.					
3. Develop and manage modules using built-in and custom Node.js modules					
4. Use NPM effectively to install, update, and manage project dependencies.					
5. Create web servers and handle HTTP requests and responses using Node.js.					
6. Perform file system operations such as reading, writing, updating, and deleting files					
7. Implement event-driven programming using the Event Emitter class					
8. Connect Node.js applications to databases and perform basic database operations like select, update, and delete.					
Chapter 1 Introduction to Node.js (4 Hours)					
Objectives: To Understand the JavaScript and technical concepts behind Node JS					
Topics Covered:					
1.1 Introduction					
1.2 What is Node.js?					
1.3 Advantages of Node.js					
1.4 Traditional Web Server Model					
1.5 Node.js Process Model					
1.6 Installing Node.js on Windows					
1.7 Working in Node.js REPL					

Chapter 2 : Node.js Modules & Core Features (7 Hours)
Topics Covered: 2.1 Functions in Node.js 2.2 Buffer 2.3 Module system 2.4 Module types 2.5 Core modules 2.6 Local modules 2.7 module.exports
Chapter 3 : NPM, Web Server & File System (9 Hours)
Topics Covered: 3.1 Node Package Manager (NPM) 3.1.1 What is NPM? 3.1.2 Installing packages locally 3.1.3 Adding dependencies in package.json 3.1.4 Installing packages globally 3.1.5 Updating packages 3.2 Web Server in Node.js 3.2.1 Creating a web server 3.2.2 Handling HTTP requests 3.2.3 Sending responses 3.3 File System (FS Module) 3.3.1 fs.readFile 3.3.2 Writing a file 3.3.3 Writing a file asynchronously 3.3.4 Opening a file 3.3.5 Deleting a file 3.3.6 Other I/O operations
Chapter 4 : Events & Database Connectivity (10 Hours)
Objectives: Build a Web Server in Node and understand how it really works , Connect to a SQL or Mongo database in Node, Build a Web Server in Node and understand how it really Works
Topics Covered: 4.1 Events in Node.js 4.1.1 Event Emitter class 4.1.2 Returning event emitters 4.1.3 Inheriting events 4.2 Database Connectivity 4.2.1 Connection string 4.2.2 Database configuration 4.2.3 Working with SELECT command 4.2.4 Updating records 4.2.5 Deleting records
Teaching Methodology:
Lecture, Discussion, Problem Solving

Assessment & Evaluation:
Internal: 20
Internal Exam :10 + Assignment / OBT / Viva :10 External :30
Bibliography:
1. Reference Books: 1) Node.js complete reference guid , velentinBojinov, David Herron, DiogeResende, packt Publishing ltd
2. Mastering Nod.js By SandroPasquali , packt Publishing
3. Smashing Node.js Javascript Everywhere , Guillermo Rauch, John wiley& Sons

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-354-MJ-TH	Compiler Construction	2	TH	CIE:20 SEE:30
Course Hours: 30					
Course Objectives:					
1. To understand design issues of a lexical analyzer and use of LEX tool..					
2. To understand design issues of a parser and use of YACC tool.					
3. To understand and design code generation and optimization techniques					
Course Outcomes: (COs)					
On completion of the course, student will be able to–					
1. Understand the process of scanning and parsing of source code.					
2. Learn the conversion code written in source language to machine language.					
Syllabus:					
Chapter 1 : Introduction (2 Hours)					
Topics Covered:					
1.1 Definition of Compiler, Aspects of compilation. The structure of Compiler.					
1.2 Phases of Compiler – Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code generation, code optimization, code generation. Error Handling.					
1.3 Cross compiler, Bootstrapping.					
Chapter 2 : Lexical Analysis (Scanner) (2 Hours)					
Topics Covered:					
2.1 Review of Finite automata as a lexical analyzer, Applications of Regular Expressions and Finite Automata (lexical analyzer, searching using RE), Input buffering, Recognition of tokens.					
2.2 LEX: A Lexical analyzer generator (Simple Lex Program)					
Chapter 3 : Syntax Analysis (Parser) (18 Hours)					
Topics Covered:					
3.1 Definition, Types of Parsers Top-Down Parser – Top-Down Parsing with Backtracking: Method & Problems Drawbacks of Top-Down parsing with backtracking					
3.2 Elimination of Left Recursion (direct & indirect)					
3.3 Need for Left Factoring & examples Recursive Descent Parsing: Definition Implementation of Recursive Descent Parser Using Recursive Procedures					
3.4 Predictive [LL (1)] Parser (Definition, Model)					
3.4.1 Implementation of Predictive Parser [LL (1)]					
3.4.2 FIRST & FOLLOW Construction of LL (1) Parsing Table Parsing of a String using LL (1) Table. 3.5 Bottom-Up Parsers					
3.5.1. Operator Precedence Parser -Basic Concepts Operator Precedence Relations f form Associativity & Precedence					
3.5.2. Operator Precedence Grammar Algorithm for LEADING & TRAILING (with ex.)					
3.5.3. Algorithm for Operator Precedence Parsing (with ex.)					
3.5.4. Precedence Functions Shift Reduce Parser Reduction, Handle, Handle Pruning					
3.5.5. Stack Implementation of Shift Reduce Parser (with examples) LR Parser: Model					

3.4.3 Types [SLR (1), Canonical LR, LALR]-Method & examples.
Chapter 4 : Syntax Directed Definition (4 Hours)
Topics Covered: 4.1 Syntax Directed Definitions (SDD) Inherited & Synthesized Attributes Evaluating an SDD at the nodes of a Parse Tree, Example Evaluation Orders for SDD's 4.2 Dependency Graph Ordering the Evaluation of Attributes S-Attributed Definition L-Attributed Definition Application of SDT Construction of syntax trees
Chapter 5 : Code Generation and Optimization (4 Hours)
Topics Covered: 5.1 Compilation of expression – Concepts of operand descriptors and register descriptors with example. 5.2 Intermediate code for expressions – postfix notations, Triples, Quadruples and Expression trees. 5.3 Code Optimization – Optimizing transformations – compile time evaluation, elimination of common sub expressions, dead code elimination, frequency reduction, strength reduction. 5.4 Three address code DAG for Three address code The Value-number method for constructing DAG's. Definition of basic block and flow graphs Directed acyclic graph (DAG).
Teaching Methodology: Lecture, Problem Solving, Discussion, Class Test, Revision
Assessment & Evaluation:
Internal Evaluation :- 20 Marks
External Evaluation:- 30 Marks
Bibliography:
1. Compilers: Principles, Techniques, and Tools, Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, 2004 2. Principles of Compiler Design By: Alfred V. Aho, Jeffrey D. Ullman, Narosa Publication House, 2002 3. LEX & YACC, 2nd edition, O'reilly Publication, 2012 <i>s with Authors</i>

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-355-MJ-PR	Lab Course based on CS-351-MJ-TH	2	PR	CIE-20 SEE-30
Course Hours: 60					
Course Objectives:					
1. To learn database programming using Java					
2. To study web development concept using Servlet and JSP					
3. To develop applications using multithreading					
4. To implement various Collections.					
Course Outcomes: (COs)					
On completion of the course, student will be able to–					
1. To access open database through Java programs using Java Data Base Connectivity (JDBC) and develop the application					
2. Understand and create dynamic web pages, using Servlets and JSP.					
3. Work with basics of framework to develop secure web applications					
4. Understand the use of collections in various applications.					
Syllabus:					
Assignment 1: Java Database Programming-JDBC					
Topics Covered:					
1. To communicate with a database using java.					
2. To execute queries on tables.					
3. To obtain information about the database and tables.					
Assignment 2 : Collections					
Topics Covered:					
1. Study the Collection framework in java.					
2. To Implement various Interfaces and classes through algorithms.					
3. To Demonstrate Cursor Objects (Enumeration, Iterator, ListIterator, Comparator)					
Assignment 3: Multithreading					
Topics Covered:					
1. To create and use threads in java.					
2. To demonstrate multithreading using Thread Synchronization,					
3. Inter-thread . Communication,					
4. Thread Priorities.					
Assignment 4: Servlet					
Topics Covered:					
1. To understand server-side programming.					
2. Simple steps to create and execute servlets.					
3. How to pass parameters using doGet and doPost methods.					
4. Handling data from HTML to servlet .					
5. How to connect servlet to a database .					
6. Use of various session tracking methods like Cookies					

Assignment 5: JSP
Topics Covered: 1. JSP life-cycle. Use of JSP implicit objects. JSP Directives. 2. Use of Scripting Elements. 3. To understand actiontags in JSP. Understanding flow of JSP custom tags
Assignment 6: Spring
Topics Covered: To create and understand the steps to develop Spring application.
Teaching Methodology:
Lectures, You-tube link sharing and discussion on related topic
Assessment & Evaluation:
Assignment, Viva
Bibliography:
1. Java 2 programming black books, Steven Horlizer
2. Herbert Schildt, "Complete reference Java (5th edition)", Tata McGraw Hill Edition, 2005
3. Getting started with Spring Framework: covers Spring 5 by J Sharma and Ashish Sarin
Webliography:
1. https://www.tutorialspoint.com
2. http://www.javatpoint.com
3. https://spring.io/projects/spring-framework
4. https://docs.oracle.com/javase/tutorial/

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-356-MJ-PR	Lab Course based on CS-352-MJ-TH	2	PR	CIE:20 SEE:30
Course Hours: 60					
Prerequisites:					
6. Basic Web Technologies • Understanding of HTML and CSS for creating static web pages • Familiarity with basic web page structure and styling 7. Fundamentals of Programming • Knowledge of variables, data types, operators, and control structures • Basic understanding of functions and logical flow 8. Introduction to Client–Server Architecture • Basic idea of how web browsers and servers communicate • Awareness of HTTP request–response mechanism 9. Basic Database Concepts • Understanding of data storage, tables, and records • Familiarity with basic SQL operations 10. Problem-Solving Skills • Ability to analyse real-world problems and implement logical solutions • Willingness to learn and apply new web development tools					
Course Objectives:					
1. To understand state management in web applications ○ To learn the concepts of cookies and sessions ○ To manage user data across multiple web pages securely 2. To explore structured data representation ○ To create and validate XML documents ○ To manipulate XML data using DOM techniques 3. To develop client-side scripting skills ○ To implement interactive web pages using JavaScript ○ To simplify DOM manipulation using jQuery 4. To implement asynchronous web communication ○ To understand AJAX concepts ○ To develop responsive web applications with partial page updates 5. To introduce framework-based development ○ To understand MVC architecture ○ To develop scalable web applications using CodeIgniter 6. To integrate front-end and back-end technologies ○ To combine JavaScript, AJAX, PHP, databases, and frameworks ○ To build complete web-based solutions 7. To enhance professional development skills ○ To write clean, reusable, and well-documented code ○ To prepare structured lab records and technical documentation					

Course Outcomes: (COs)
CO1: Recall and explain the fundamentals of state management, XML documents, client- side scripting, AJAX, and PHP frameworks and differentiate between cookies and sessions, understand XML structure, and MVC architecture in CodeIgniter. CO2: Apply PHP to implement cookies and sessions for maintaining user state in web Applications and use DOM techniques to parse and manipulate XML data effectively. CO3: Write JavaScript and jQuery code to handle user interactions, validate inputs, and manipulate web page elements and implement AJAX for asynchronous client-server communication and partial page updates. CO4: Analyze web application requirements and decide when to use cookies, sessions, XML parsing, or AJAX for better performance and evaluate JavaScript/jQuery and AJAX implementations for correctness, efficiency, and user experience. CO5: Develop web applications using the CodeIgniter framework, implementing MVC architecture and CRUD operations and integrate client-side (JavaScript/jQuery, AJAX) and server-side (PHP, CodeIgniter) technologies to create cohesive web applications. CO6: Design, test, and debug complete web solutions that combine cookies/sessions, XML, AJAX, and MVC framework concepts and assess web applications for ability, maintainability, and performance according to modern web standards.
Syllabus:
Assignment 1: Cookies and Sessions
Topics Covered: • Introduction to state management in web applications • Concept of cookies and sessions • Creating and retrieving cookies • Setting cookie expiry time • Creating and managing sessions • Storing and accessing session variables • Difference between cookies and sessions • Simple programs demonstrating cookie and session usage
Assignment 2: XML Documents and DOM
Topics Covered: • Introduction to XML and its features • Structure and syntax of XML documents • XML elements, attributes, and namespaces • XML validation using DTD/XSD (basic concept) • Introduction to DOM (Document Object Model) • Parsing XML using DOM • Accessing and modifying XML elements using DOM methods
Assignment 3: JavaScript and jQuery
Topics Covered: • Introduction to JavaScript and client-side scripting • JavaScript variables, operators, and functions • Conditional statements and loops in JavaScript • DOM manipulation using JavaScript • Form validation using JavaScript • Introduction to jQuery library • jQuery selectors and events • DOM manipulation using jQuery

Assignment 4: AJAX
Topics Covered: • Introduction to AJAX (Asynchronous JavaScript and XML) • Concept of asynchronous communication • AJAX architecture and workflow • Using XMLHttpRequest object • Sending and receiving data using AJAX • Partial page updates without page reload • Implementing AJAX using JavaScript/jQuery
Assignment 5: PHP Framework – CodeIgniter
Topics Covered: • Introduction to PHP frameworks • Overview of CodeIgniter framework • Installation and configuration of CodeIgniter • MVC architecture (Model–View–Controller) • Creating controllers, models, and views • Routing and basic configuration • Database connectivity in CodeIgniter • Implementing CRUD operations using MVC
Teaching Methodology:
1. Conceptual Explanation 2. Demonstration Method 3. Hands-on Practice 4. Step-by-Step Guidance 5. Assignment-Based Learning 6. Discussion and Doubt Solving 7. Continuous Evaluation
Assessment & Evaluation:
1. As mentioned above in the Evaluation and Assessment section.
Bibliography:
1. Flanagan, D. JavaScript: The Definitive Guide, 7th Edition, O'Reilly Media, 2020. 2. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development, John Wiley & Sons, 2014. 3. Deitel, P., & Deitel, H. Internet and World Wide Web: How to Program, 5th Edition, Pearson Education, 2012. 4. Holzner, S. Ajax: A Beginner's Guide, McGraw-Hill Education, 2008. 5. McLaughlin, B. PHP and MySQL Web Development, 5th Edition, Addison-Wesley, 2016. 6. Ullman, L. PHP for the Web: Visual QuickStart Guide, Peachpit Press, 2016. 2. W3C. XML Specification and DOM Level Standards, World Wide Web Consortium, Latest Edition.

Webliography:

1. **MDN Web Docs (Mozilla Developer Network)**
<https://developer.mozilla.org>
Authoritative documentation for JavaScript, DOM, AJAX, and web standards.
2. **W3Schools**
<https://www.w3schools.com>
Tutorials and references for JavaScript, jQuery, XML, AJAX, PHP, and Cookies.
3. **jQuery Official Documentation**
<https://jquery.com>
API reference and examples for jQuery library.
4. **PHP Official Documentation**
<https://www.php.net/docs.php>
Complete reference for PHP functions, sessions, cookies, and security.
5. **CodeIgniter Official Documentation**
https://codeigniter.com/user_guide/
Guidelines for MVC architecture and framework usage.
6. **XML.com**
<https://www.xml.com> *Resources for XML structure, validation, and parsing.*

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-360-MJ-TH	Android programming	2	TH	CIE:20 SEE:30
Course Hours: 30 Hours					
Course Objectives:					
1. To understand the Android Operating System					
2. To study Android Apps Development Cycle					
3. To learn to create Android Applications					
4. To understand the basic Android UI design					
Course Outcomes: (COs)					
1. Describe the process of developing mobile applications.					
2. Create mobile applications on the Android Platform.					
3. Design and implement basic mobile applications.					
4. Developing android applications using menu.					
Syllabus:					
Chapter 1 : Introduction to Android (06 Hours)					
Topics Covered:					
1.1. Overview					
1.2. History					
1.3. Features of Android					
1.4. Architecture of Android					
1.4.1 Overview of Stack					
1.4.2 Linux Kernel					
1.4.3 Native Libraries					
1.4.4 Android Runtime					
1.4.5 Application Framework					
1.4.6 Applications					
1.5. SDK Overview					
1.5.1 Platforms					
1.5.2 Tools – (JDK, SDK, Eclipse/Android Studio, ADT,AVD, Android Emulator), Versions					
Chapter 2 : Activities, Toast and Intents (06 Hours)					
Topics Covered:					
2.1. Introduction to components of Android OS					
2.2. Introduction to Activities					
2.3. Activity Lifecycle					
2.4. Creating your first Android Application					
2.5. Introduction to Intents					
2.6. Linking Activities using Intents					
2.7. Calling built-in applications using Intents					
2.3. Toast					

Chapter 3 : Android User Interface with layouts (08 Hours)
Topics Covered: 3.1. Understanding the components of a screen 3.2. Views and View Groups 3.2.1 Linear Layout 3.2.2 Absolute Layout 3.2.3 Table Layout 3.2.4 Relative Layout 3.2.5 Frame Layout 3.2.6 Scroll View 3.2.7 Constraint Layout 3.2.8 Grid Layout 3.3. Adapting to Display Orientation 3.3.1 Anchoring Views 3.3.2 Resizing and Repositioning
Chapter 4 : Designing Your User Interface with Views (06 Hours)
Topics Covered: 4.1. Using Basic Views 4.1.1 TextView 4.1.2 Button, ImageButton, EditText, CheckBox 4.1.3 Switch, ToggleButton, RadioButton, and RadioGroup Views 4.1.4 ProgressBar View 4.1.5 AutoCompleteTextView View 4.2. Using Picker Views 4.2.1 TimePicker View 4.2.2 DatePicker View 4.3. Using List Views to Display Long Lists 4.3.1 ListView View 4.3.2 Using the Spinner View
Chapter 5 : Adapter and Menus (04 Hours)
Topics Covered: 5.1. Adapter and its types 5.1.1 Base Adapter 5.1.2 Array Adapter 5.2. Using Menus with Views 5.2.1 Options Menu 5.2.2 Context Menu 5.2.3 Popup menu
Teaching Methodology:
Lecture, Demonstration, Problem Solving
Assessment & Evaluation:
Internal: 20 marks External : 30 marks

Bibliography:
1) <i>Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication</i> 2) <i>Professional Android 4 Application Development, By Reto Meier WROX Publication</i> 3) <i>Head First Android Development: A Brain-Friendly Guide, By David Griffiths and Dawn Griffiths</i> 4) <i>Mobile App Development Technologies M.Sc.(C.S.)-Sem-II by Monica Deshmane- Dhotre , Nirali Publication(2023) .</i>
Webliography:
E-Books: 1) https://enos.itcollege.ee/~jpoial/allalaadimised/reading/Android-Programming-Cookbook.pdf 2) https://www.programming-book.com/download/?file=10988 3) http://projanco.com/Library/Android%20App%20Development%20in%20Android%20Studio%20-%20Java%20plus%20Android%20edition%20for%20beginners.pdf 4) https://www.programmer-books.com/professional-android-4th-edition-pdf/
Websites: 1) <i>The official site for Android developers - https://developer.android.com</i> 2) https://www.tutorialspoint.com/android/index.htm 3) https://www.javatpoint.com/android-tutorial

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year: 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-361-MJ-PR	Lab Course based on CS-360-MJ-TH	2	PR	CIE:20 SEE:30
Course Hours: - 60 Hours					
Course Objectives:					
1. To understand the Android Operating System					
2. To study Android Apps Development Cycle					
3. To learn to create Android Applications					
4. To understand the basic Android UI design					
Course Outcomes: (COs)					
1. Describe the process of developing mobile applications.					
2. Create mobile applications on the Android Platform.					
3. Design and implement basic mobile applications.					
4. Developing android applications using menu.					
Syllabus:					
Assignment 1 : Introduction to Android (02 Hours)					
Topics Covered:					
1.1. SDK Overview					
• Platforms • Tools – (JDK, SDK, Eclipse/Android Studio, ADT,AVD, Android Emulator), Versions • Installation demo					
Assignment 2 : Activities (02 Hours)					
Topics Covered:					
2.1. Demo for Activity Lifecycle for each state.					
2.2. Creating your first Android Application					
Assignment 3 : Toast (02 Hours) Topics Covered:					
3.1. Create application with Toast					
Activities: Create android app with toast.					
Assignment 4 : Intents (06 Hours) Topics Covered:					
4.1. Create Implicit Intents					
4.2. Linking Activities using Intents					
4.3. Calling built-in applications using Intent					
Assignment 5 : Android User Interface with layouts (08 Hours)					
Topics Covered:					
5.1 Linear Layout					
5.2. Absolute Layout					
5.3. Table Layout					
5.4. Relative Layout					

Assignment 6 : Android User Interface with layouts (08 Hours) Topics Covered: 6.1. Frame Layout 6.2. Scroll View 6.3. Constraint Layout 6.4. Grid Layout
Assignment 7 : Designing Your User Interface with Views (08 Hours) Topics Covered: 7.1. Using Basic Views • TextView • Button, ImageButton, EditText, CheckBox • Switch, ToggleButton, RadioButton, and RadioGroup Views • ProgressBar View • AutoCompleteTextView View 7.2. Using Picker Views • TimePicker View • DatePicker View
Assignment 8 : Designing Your User Interface with Views (08 Hours) Topics Covered: 8.1. Using List Views to Display Long Lists • ListView View • Using the Spinner View
Assignment 9 : Adapter (08 Hours) Topics Covered: 9.1. Adapter and its types • Base Adapter • Array Adapter
Assignment 10 : Menus (08 Hours) Topics Covered: 10.1. Using Menus with Views • Options Menu • Context Menu • Popup menu
Teaching Methodology: Explanation, Demo, Discussion
Assessment & Evaluation: Internal: 20 marks External : 30 marks
Bibliography: 1) <i>Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication</i> 2) <i>Professional Android 4 Application Development, By Reto Meier WROX Publication</i> 3) <i>Head First Android Development: A Brain-Friendly Guide, By David Griffiths and Dawn Griffiths</i> 4) <i>Mobile App Development Technologies M.Sc.(C.S.)-Sem-II by Monica Deshmane- Dhotre , Nirali Publication(2023)</i>

Webliography:*E-Books:*

- 1) <https://enos.itcollege.ee/~jpoial/allalaadimised/reading/Android-Programming-Cookbook.pdf>
- 2) <https://www.programming-book.com/download/?file=10988>
- 3) <http://projanco.com/Library/Android%20App%20Development%20in%20Android%20Studio%20-%20Java%20plus%20Android%20edition%20for%20beginners.pdf>
- 4) <https://www.programmer-books.com/professional-android-4th-edition-pdf/>

Websites:

- 1) The official site for Android developers - <https://developer.android.com>
- 2) <https://www.tutorialspoint.com/android/index.htm>
- 3) <https://www.javatpoint.com/android-tutorial>

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-362-MJ-TH	Software Testing Tools	2	TH	CIE-20 SEE-30
Course Hours: 30					
Prerequisites:					
• Basic knowledge of algorithms, problem solving, expected inputs/outputs • Knowledge of C and Java Programming Language, compilation, debugging					
Course Objectives:					
1. To provide the knowledge of software testing methods and strategies.					
2. To understand how testing methods can be used as an effective tool in quality assurance of software.					
3. To provide skills to design test case plan for testing software.					
4. To provide knowledge of latest testing methods.					
5. To provide knowledge of latest testing tools.					
Course Outcomes: (COs)					
1. To understand various software testing methods and strategies.					
2. To understand a variety of software metrics and identify defects and managing those Defects for improvement in quality for given software.					
3. To design test cases and test plans, review reports of testing for qualitative software.					
4. To understand latest testing methods used in the software industries					
5. To understand latest testing tools used in the software industries.					
Syllabus:					
Chapter 1: Introduction to Software Testing (5 hours)					
Topics Covered: 1.1 Basics of Software Testing – faults, errors and failures 1.2 Testing objectives 1.3 Principles of testing 1.4 Testing and debugging 1.5 Testing metrics and measurements 1.6 Verification and Validation 1.7 Testing Life Cycle					
Chapter 2: Introduction to Test case design (8 hours)					
Topics Covered: 2.1 Basics for Test Case design 2.1.1 White box Testing, Black box Testing. 2.1.2 Difference between BBT and WBT 2.2 How to identify errors, bugs in the given application. 2.3 Design entry and exit criteria for test case, 2.4 Design test cases in excel. 2.2 Describe feature of a testing method used.					

Chapter 3 : Test cases for simple programs (5 hours)
Topics Covered: 3.1 Write simple programs make use of loops and control structures. 3.2 Write Test Cases for above programs.
Chapter 4 : Test cases and Test plan (6 hours)
Topics Covered: 4.1 Write Test Plan for given application with resources required. 4.2 Write Test case for given application. 4.3 Prepare Test report for test cases executed.
Chapter 5: Defect Report (3 hours)
Topics Covered: 5.1 Defect Life Cycle 5.2 Classification of Defect 5.3 Write Defect Report
Chapter 6: Testing Tools (3 hours)
Topics Covered: 6.1 How to make use of Automation Tools 6.2 Types of Testing Tools
Teaching Methodology: 1. Lecture Method 2. Demonstration-Based Learning 3. Problem-Solving Approach 4. Activity-Based Learning
Assessment & Evaluation: As mentioned above in the Evaluation and Assessment section.
Bibliography: 1. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill, 20 2. Effective Methods of Software Testing, William E Perry, 3rd Edition, Wiley Publishing Inc 3. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing, Rex Black, Microsoft Press, 1999 4. Software Testing Principles and Practices by Srinivasan Desikan, Gopalaswamy Ramesh, Pearson
Webliography:

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-363-MJ-PR	Lab Course based on CS-362MJ-TH	2	PR	CIE-20 SEE-30
Course Hours: 30					
Prerequisites:					
- Basic knowledge of algorithms, problem solving, expected inputs/outputs - Knowledge of C and Java Programming Language, compilation, debugging					
Course Objectives:					
1. To understand the fundamental concepts of software testing and quality assurance.					
2. To learn different types of testing such as manual testing and automation testing.					
3. To design and write effective test cases for software applications.					
4. To understand test planning, test execution, and test reporting.					
5. Identify, document, and report software defects accurately.					
6. Prepare test plans, test cases, and test reports as per industry standards.					
Course Outcomes: (COs)					
1. Understand and apply basic software testing concepts in real-world scenarios.					
2. Design and execute manual test cases effectively.					
3. Use automation testing tools like Selenium for web application testing.					
4. Perform browser automation and validate web application behaviour.					
5. Demonstrate practical skills required for entry-level software testing roles.					
Syllabus:					
Assignment 1 : Introduction to Test Case Design					
Tool/Platform: Excel/Google Sheets or Word.					
Procedure:					
- Given a simple application (like a calculator), identify possible input errors. - Write test cases using columns: Test ID, Description, Input, Expected Output, Actual Output, Status (Pass/Fail). - Use Excel to record test cases.					
Output: Table of test cases.					
Conclusion: Discuss which bugs were found and how test cases helped.					
Assignment 2 : Test Cases for Simple Programs					
Objectives: Write test cases for simple programs (loops, control structure).					
Procedure:					
- Take a C/Java program that finds factorial or sum of series. - List possible valid and invalid inputs. - Write test cases in Lab Book.					
Output: Completed test cases with expected and actual results.					
Conclusion: Reflect on how thorough your test cases were.					

Assignment 3: Test Plan & Test Report
Objectives: Prepare a Test Plan and Test Report for an application.
Procedure: 1. Write a concise Test Plan for a mini project (like login feature). 2. Include scope, resources, schedule, entry/exit criteria. 3. Execute test cases and fill Test Report fields. Output: Test Plan Document and Test Report. Conclusion: Check effectiveness of testing.
Assignment 4 : Defect Report and Defect Life Cycle
Procedure: 1. While testing, identify defects/bugs. 2. For each defect, write steps to reproduce, severity, priority, expected & actual behavior. 3. Show Defect Life Cycle transitions (Open → Assigned → Fixed → Closed). Output: Defect Report Table. Conclusion: Learning on defect documentation.
Assignment 5 : Using Automation Testing Tools
Objectives: Demonstrate basic usage of a testing tool (e.g., Selenium WebDriver, Bugzilla, JMeter – open source options recommended).
Tool: • Selenium (browser automation) • Bugzilla (defect tracking) • JMeter (load testing) Procedure (for Selenium example): 1. Install Selenium WebDriver and required language bindings. 2. Write a simple script to open a website and check title. 3. Capture screenshots and logs. Output: Code, execution result, screenshot. Conclusion: Advantages of automation compared to manual testing.
Teaching Methodology:
1. Conceptual Explanation 2. Demonstration Method 3. Hands-on Practice 4. Step-by-Step Guidance 5. Assignment-Based Learning 6. Discussion and Doubt Solving 7. Continuous Evaluation
Assessment & Evaluation:
As mentioned above in the Evaluation and Assessment section.
Bibliography:
1. Pressman, R. S., <i>Software Engineering – A Practitioner’s Approach</i>, McGraw-Hill Education. 2. Perry, William E., <i>Effective Methods for Software Testing</i>, John Wiley & Sons. 3. Black, Rex, <i>Managing the Testing Process</i>, Wiley Publishing. 4. Myers, Glenford J., Sandler, Corey, and Badgett, Tom, <i>The Art of Software Testing</i>, Wiley. 5. Sommerville, Ian, <i>Software Engineering</i>, Pearson Education.

Webliography:

1. Selenium Official Documentation – <https://www.selenium.dev>
1. Oracle Java Documentation – <https://docs.oracle.com/javase>
2. Eclipse IDE Documentation – <https://www.eclipse.org/documentation>
3. Software Testing Help Tutorials – <https://www.softwaretestinghelp.com>
4. Apache JMeter Documentation – <https://jmeter.apache.org>
5. Bugzilla Official Website – <https://www.bugzilla.org>

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-371-VSC-PR	Lab Course based on CS-353-MJTH	2	PR	CIE:20 SEE:30
Course Hours: 60					
Course Objectives:					
1.Introduce students to the fundamentals of Node.js and its role in modern server-side web development.					
2.Explain the Node.js runtime environment , event-driven architecture, and non-blocking I/O model.					
3.Familiarize students with Node.js modules , core libraries, and package management using NPM .					
4.Enable students to develop web servers and handle HTTP requests and responses using Node.js.					
5.Provide hands-on knowledge of file system operations , event handling, and asynchronous Programming					
6. De.velop skills to connect Node.js applications with databases and perform basic CRUD operations					
Course Outcomes: (COs)					
1.Understand and explain the architecture, features, and advantages of Node.js.					
2.Install and configure the Node.js environment and use REPL for testing code.					
3.Develop and manage modules using built-in and custom Node.js modules					
4.Use NPM effectively to install, update, and manage project dependencies.					
5.Create web servers and handle HTTP requests and responses using Node.js.					
6.Perform file system operations such as reading, writing, updating, and deleting files					
7.Implement event-driven programming using the EventEmitter class					
8.Connect Node.js applications to databases and perform basic database operations like select, update, and delete.					
Assignment 1 : Introduction to Node.js & REPL					
- Objectives: Understand Node.js environment - Work with Node.js REPL					
Topics Covered:					
Assignment 2 : Node.js Modules					
Objectives: Learn module creation and usage					
Topics Covered:					
Install Node.js on Windows					
Use Node.js REP					

Assignment 3 : Node .Js Function
Objectives: Learn function creation and usage
Topics Covered:
1 Demonstrate the use of function.
Assignment 4 : Buffer & File Handling Basics
Objectives: Understand buffers and basic file operations
Topics Covered:
1. Create a buffer and write data into it.
2. Read data from the buffer and display it.
3. Create a text file and write content synchronously.
4. Read the file using fs.readFile().
Assignment 5 : Node Package Manager (NPM)
Objectives: Learn package management using NPM
Topics Covered:
1. Initialize a Node.js
2. Install any package locally (example: lodash)
3. Add dependency in package.json
4. Update the installed package.
Assignment 6 : Creating a Web Server
Objectives: Create and run a web server
Topics Covered:
1. Create a simple HTTP web server using Node.js.
2. Handle different URL requests:
Assignment 7: Event Handling in Node.js
Objectives: Implement event-driven programming
Topics Covered:
1. Create an event using Event Emitter.
2. Emit an event and handle it with a listener.
3. Pass arguments while emitting events.
4. Demonstrate event inheritance.
Assignment 8: Database Connectivity
Objectives: Connect Node.js with database
Topic Covered:
2. Connect Node.js application to a database (MySQL / MongoDB).
3. Create a table / collection.
4. Insert sample records.
5. Perform:
o SELECT operation
o UPDATE operation
o DELETE operation
6.. Display results on console or browser
Teaching Methodology:
Lecture, Problem Solving, Demonstration

Assessment & Evaluation:
Internal: 20 Internal Exam :10 + Assignment / OBT / Viva :10 External :30
Bibliography:
1. Reference Books: 1) Node.js complete reference guid , velentinBojinov, David Herron, DiogeResende, packt Publishing ltd
2. Mastering Nod.js By SandroPasquali , packt Publishing
3. Smashing Node.js Javascript Everywhere , Guillermo Rauch, John wiley& Sons

Restructured Syllabus (as per NEP 2024 Pattern)

To be implemented from Academic Year : 2026-27

Semester	Course Code	Course Title	Credits	Type (TH/PR)	Marks
VI	CS-381-OJT-PR	On Job Training	4	PR	CIE-40 SEE-60
Course Hours: 120					
Course Objectives:					
1. Apply theoretical computer science knowledge in real-world work environments.					
2. Develop software solutions using appropriate tools and technologies.					
3. Demonstrate professional ethics, teamwork, and communication skills.					
4. Analyse problems and propose effective technical solutions.					
5. Prepare technical documentation and present project outcomes.					
Course Outcomes: (COs)					
1. Apply concepts in real projects.					
2. Develop and test applications.					
3. Demonstrate professional skills.					
4. Solve industry-based problems.					
5. Document and present work.					
OJT Guidelines for Students:					
1. Eligibility & Duration - Minimum 120 hours of OJT is compulsory. - OJT must be completed in a recognized organization with prior college approval.					
2. Before Joining OJT - Obtain approval from the OJT Coordinator / Internal Mentor. - Submit organization acceptance letter. - Understand working hours, rules, and assigned supervisor.					
3. During OJT - Maintain discipline, punctuality, and professionalism. - Follow organization rules and confidentiality norms. - Maintain a weekly logbook signed by the Internal Mentor. - Communicate regularly with the internal mentor.					
4. After Completion of OJT - Obtain completion certificate and evaluation form from organization. - Submit OJT report, logbook, and certificates within deadline. - Attend presentation and viva-voce.					
5. Code of Conduct - Misconduct, plagiarism, or false reporting will lead to disqualification. - Students are responsible for their own conduct and safety during OJT.					
6. Mandatory Documents to submit - Offer Letter - Relieving Letter by College - Completion certificate					

Teaching Methodology:
As mentioned above in the instruction section
Assessment & Evaluation:
As mentioned above in the Evaluation and Assessment section.

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

Eligibility:

1. H.S.C. (10 + 2) Science stream with Mathematics.
2. Three years diploma course after S.S.C. (10th std.) of Board of Technical Education conducted by Government of Maharashtra or its equivalent.

Objectives:

- To develop problem solving abilities using a computer.
- To build the necessary skillset and analytical abilities for developing computer-based solutions for real
- life problems.
- To train students in professional skills related to the Software Industry.
- To prepare the necessary knowledge base for research and development in Computer Science.
- To help student's build-up a successful career in Computer Science and to produce entrepreneurs who can
- Innovate and develop software products.

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

Workload

1. Each theory credit is equivalent to 15 clock hours of teaching (i.e. for 2 Credits – 30 Clock Hours) and each practical credit is equivalent to 30 clock hours (i.e. for 2 Credits – 60 Clock Hours) of teaching in a semester.
2. There is 15 weeks of teacher-student interaction during the semester.
3. The 15 week is divided into 12 weeks teaching and 3 weeks for continuous assessment including preparation time to students during the semester.
4. The workload will be calculated based on 12 weeks teaching only.
5. For the purpose of computation of work-load the following mechanism may be adopted as per UGC guidelines.
6. Workload as per credit is as follows:
 - i. 1 Credit = 1 Theory period of one hour duration per week.
 - ii. 1 Credit = 1 Tutorial period of one hour duration per week.
 - iii. 1 Credit = 1 Practical period of two-hour duration per week.
7. Each theory Lecture time for FY, SY, TY is of 60 min. (2 Lectures per week for 2 credit courses)
8. Each practical session time for FY/ SY /TY is of 4 hour i.e. 240 min.

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
 (Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

Credit Framework

Level /Difficulty	Sem	Subject-1				Subject-2	Subject-3	GE/ OE	SEC	IKS	AEC	VEC	CC	Total
4.5/100	I	2(T)+2(P)				2(T)+2(P)	2(T)+2(P)	2(T)	2(T/P)	2(T) Generic	2(T)	2	-	22
	II	2(T)+2(P)				2(T)+2(P)	2(T)+2(P)	2(P)	2(T/P)	-	2(T)	2	2	22
Exit Option: Award of UG <i>Certificate</i> in Major with 44 credits and an additional 4 credits core NSQF Course/Internship OR Continue with Major and Minor Continue Option: Student will select one subject among the (Subject-1, Subject-2, and Subject-3) as major and another as minor and third subject will be dropped.														
Level /Difficulty	Sem	Credits related to major				Minor		GE/ OE	SEC	IKS	AEC	VEC	CC	Total
		Discipline Specific Core (DSC) Major Core	Discipline Specific Elective (DSE) Major Elective	VSC	FP/ OJT /CEP									
5.0/200	III	4(T)+2(P)	-	2(T/P)	2 (FP)	2(T)+2(P)	-	2(T)	-	2(T)	2(T)	-	2	22
	IV	4(T)+2(P)	-	2(T/P)	2(CEP)	2(T)+2(P)	-	2(P)	2(T/P)	-	2(T)	-	2	22
Exit Option: Award of UG <i>Diploma</i> in Major and Minor with 88 credits and an additional 4 credits core NSQF Course/Internship OR Continue with Major and Minor.														
5.5/300	V	8(T)+4(P)	2(T)+2(P)	2(T/P)	(FP/ CEP)	2(T)	-	-	-	-	-	-	-	22
	VI	8(T)+4(P)	2(T)+2(P)	2(T/P)	4(OJT)	-	-	-	-	-	-	-	-	22
Total 3 Years		44	8	8	10	18	8	8	6	4	8	4	6	132
Exit Option: Award of UG <i>degree</i> in Major with 132 credits OR Continue with Major and Minor.														
6.0/400	VII	6(T)+4(P)	2(T)+2(P)	-	4(RP)	4(T)(RM)	-	-	-	-	-	-	-	22
	VIII	6(T)+4(P)	2(T)+2(P)	-	8(RP)	-	-	-	-	-	-	-	-	22
Total 4 Years		64	16	8	22	22	8	8	6	4	8	4	6	176
Exit Option: Award of UG <i>Honors with Research</i> Degree in Major and Minor with 176 credits. -OR-														
6.0/400	VII	10(T)+4(P)	2(T)+2(P)	-	-	4(T)(RM)	-	-	-	-	-	-	-	22
	VII I	10(T)+4(P)	2(T)+2(P)	-	4(OJT)	-	-	-	-	-	-	-	-	22
Total 4 Years		72	16	8	14	22	8	8	6	4	8	4	6	132
Exit Option: Award of UG <i>Honors Degree</i> in Major and Minor with 176 credits.														

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

Reference Books

1. Java and Advanced Java

1. *E. Balagurusamy, Programming with Java, A primer, Fourth edition, Tata McGraw Hill Education Private Limited, 2009*
2. *Herbert Schildt, "Complete reference Java (5th edition)", Tata McGraw Hill Edition, 2005*
3. *Java 2 programming black books, Steven Horlizer*

2. Web Technology -I & II

1. *HTML & CSS: The Complete Reference, Fifth Edition Author: Thomas A. Powell First published: 01 Jan 2010.*
2. *Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication*
3. *Beginning PHP 5 , Wrox publication*
4. *PHP web services, Wrox publication*
5. *Mastering PHP , BPB Publication*
6. *PHP cookbook, O'Reilly publication*
7. *AJAX Black Book Kogent solution*
8. *Professional Codeigniter By Thomas Myer ,Wrox Publication*
9. *Codeigniter 2 CookBook By Rob Foster ,PACKT Publication , JQuery CookBook, O'reilly Publication*

3. Operating System

1. *Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia*
2. *Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India.*
3. *Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001*
4. *The 'C' Odyssey, UNIX-the open boundless C, Meeta Gandhi, Tilak Shetty, Rajiv Shah, BPB publication*
5. *Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001 (Text Book)*

4. Theoretical Computer Science

1. *Introduction to Automata Theory, Languages and Computation, John E. Hopcraft, Rajeev Motwani, Jeffrey D. Ullman, Third Edition, Pearson Education Publication, 2008*
2. *Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House, 1995*
3. *Theory of Computer Science Automata, Languages and Computation, K.L.P. Mishra, N. Chandrasekaran, Publication- Prentice Hall of India, 2008*
4. *Introduction to Computer Theory Daniel I. A. Cohen – 2nd edition – John Wiley & Sons, 1996 5.*

5. *Introduction to Languages and The Theory of Computation* John C. Martin The McGraw- Hill, Fourth Edition, 2011

5. Data Science

1. *Data Science Fundamentals and Practical Approaches*, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020.
2. *The Data Science Handbook*, Field Cady, John Wiley & Sons, Inc, 2017
3. *Data Mining Concepts and Techniques*, Third Edition, Jiawei Han, Micheline Kamber
4. *A Hands-On Introduction to Data Science*, Chirag Shah, University of Washington Cambridge University Press

6. Database Technology

1. *MongoDB: The Definitive Guide* by Kristina Chodorow
2. *NoSQL Distilled* by Martin Fowler
3. *MongoDB Basics* by David Hows, Peter Membrey and Eelco Plugge *NoSQL with MongoDB in 24 Hours*, Sams Teach Yourself

7. Design Framework

1. *Reference Books: 1) Node.js complete reference guid* , velentinBojinov, David Herron, DiogeResende, packt Publishing ltd
2. *Mastering Nod.js* By SandroPasquali , packt Publishing
3. *Smashing Node.js Javascript Everywhere* , Guillermo Rauch, John wiley& Sons

8. Compiler Construction

1. *Compilers: Principles, Techniques, and Tools*, Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, 2004
2. *Principles of Compiler Design* By: Alfred V. Aho, Jeffrey D. Ullman, Narosa Publication House, 2002
3. *LEX & YACC*, 2nd edition, O'reilly Publication, 2012

9. Android Programming

1. <https://enos.itcollege.ee/~jpoial/allalaadimised/reading/Android-Programming-Cookbook.pdf>
2. <https://www.programming-book.com/download/?file=10988>
3. <http://projanco.com/Library/Android%20App%20Development%20in%20Android%20Studio%20-%20Java%20plus%20Android%20edition%20for%20beginners.pdf>
4. <https://www.programmer-books.com/professional-android-4th-edition-pdf/>

10. Software Testing Tools

1. *Software Engineering – A Practitioners Approach*, Roger S. Pressman, 7thEdition, Tata McGraw Hill, 20
2. *Effective Methods of Software Testing*, William E Perry, 3rd Edition, Wiley Publishing Inc
3. *Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing*, Rex Black, Microsoft Press, 1999
4. *Software Testing Principles and Practices* by Srinivasan Desikan, Gopalaswamy Ramesh, Pearson

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

Examination Pattern

1. Exam pattern is 60-40 i.e. Semester End Examination (SEE) is of 60 % and Continuous Internal Assessment is of 40 %.

Theory, Practical/Project: -

Continuous Internal Assessment (CIA): 40 % [20 Marks]

1. Written Test- 50 % [10 Marks]
2. Presentation/Assignment: - 25% [5 Marks]
3. Project/Activity Oriented Learning: -25 % [05 Marks]

Semester End Examination (SEE): - 60 % [30 Marks]

2. For Internal examination, written test should be conducted of 10 marks for 2 credits.

Paper Pattern

Internal Examination

Que-1: Answer the following questions (Short answer questions) (any 4 out of 5) – 4 Marks.

Que-2: Answer the following questions (Short answer questions) (any 2 out of 3)- 4 Marks

Que-3: Answer the following questions (Logical Question) (any 1 out of 2)– 2 Marks.

External Examination

Que-1: [06 Marks]

Answer the following questions (Short answer/Definition/Problems/Diagram, etc.) (Any 6 out of 8)

Que-2: [08 Marks]

A] Answer the following questions (Short answer questions- 2 Que out of 3) [6 Marks]

B] Answer the following questions (Long answer questions/Problems- 1 Que out of 2) [2 Marks]

Que-3: [08 Marks]

A] Answer the following questions (Short answer questions- 1 Que out of 2 [3 Marks]

B] Answer the following questions (Long answer questions/Problems- 1 Que out of 2) [5 Marks]

Que-4: [08Marks]

A] Answer the following questions (Short answer questions- 1 Que out of 2 [3 Marks]

B] Answer the following questions (Long answer questions/Problems- 1 Que out of 2) [5 Marks]

The Poona Gujarati Kelvani Mandal's
Haribhai V. Desai College of Arts, Science and Commerce, Pune
(Autonomous)

Structure of UG Program as per NEP-2020

Program: - B.Sc. (Computer Science)

Major Course: - Computer Science

Completion of Degree

1. A student who earns 44 credits at First year with additional 4 credits core NSQF Course/Internship shall be eligible for the award of UG Certificate in Major subject. OR May continues with Major and Minor.
2. A student who earns 88 credits at Second year with additional 4 credits core NSQF Course/Internship shall be eligible for the award of UG Diploma in Major subject with Minor (Selected at SY). OR May continues with Major and Minor.
3. A student who earns 132 credits at Third year shall be eligible for the award of UG Degree in Major subject with Minor (Selected at SY). OR May continues with Major and Minor.
4. A student who earns 176 credits in Four years shall be eligible for the award of UG Honors or Honors with Research in Major subject with Minor (Selected at SY).

Award of Degree:

CGPA will be calculated for students who completed 132/176 credits, grades are given as per the following table.

Sr. No.	Grade Letter	Grade Point	Marks
1.	O (Outstanding)	10	90<= Marks <= 100
2.	A+ (Excellent)	9	75<= Marks <= 89
3.	A (Very Good)	8	60<= Marks <= 74
4.	B+ (Good)	7	55<= Marks <= 59
5.	B (Above Average)	6	50<= Marks <= 54
6.	C (Average)	5	45<= Marks <= 49
7.	D (Pass)	4	40<= Marks <= 40
8.	F (Fail)	0	Marks <40
9.	AB (Absent)	0	-

--- END ---