

Bharati Vidyapeeth
(Deemed To Be University), Pune (India)

***Accredited 'A+' Grade (2017) By NAAC ***
***'Category -I' University Status by UGC ***
*** 'A' Grade University Status by MHRD Govt. of India ***
*** Ranked '63rd' by NIRF-2020 under University Category ***

Faculty of Management Studies
Board of Studies in Computer Applications and System Studies

Master of Computer Applications Programme
(MCA)

(Under Choice Based Credit System-2026)

To be implemented from 2026-27

Content

Sr.No.	Particulars	Page No.
I	Introduction	3
II	Vision	3
III	Mission	3
IV	Programme Unique Features	3
V	Programme Objectives	4
VI	Programme Outcomes	4
VII	Programme Specific Outcomes	5
VIII	Eligibility Criteria	6
IX	Duration of the Programme	6
X	MOOC Policy	6
XI	Scheme of Examination	7
XII	Standard of Passing	7
XIII	Award of Honors	09
XIV	Question Paper Patterns for University Examination	10
XV	MCA Programme Structure	11
XVI	List of Electives	15
XVII	Bridge Course	15
	Detail Syllabus	18

Bharati Vidyapeeth (Deemed To Be University), Pune India
Faculty of Management Studies
(Board of Studies in Computer Applications and System Studies)
Master of Computer Applications Programme (2026 Course)
(Under NEP 2020)
To be effective from 2026-27 at Part I

I. INTRODUCTION:

The MCA Program is a full time 95 credits programme offered by Bharati Vidyapeeth (Deemed to be University), Pune and is conducted in regular and distance mode at its Management Institutes in Pune, Karad, Kolhapur, Sangli, and Solapur. This programme is also conducted in online mode at CDOE under BV(DU). All the five institutes have excellent faculties, laboratories, library, and other facilities to provide proper learning environment. The University is reaccredited by NAAC with an 'A++' grade (3rd cycle). The expectations and requirements of the software industry, immediately and in the near future, are visualized while designing the MCA programme. This effort is reflected in the Vision and Mission statements of the MCA programme. Of course, the statements also embody the spirit of the vision of Late Dr. Patangraoji Kadam, the Founder of Bharati Vidyapeeth and Chancellor, Bharati Vidyapeeth Deemed to be University which is to usher in “Social Transformation through Dynamic Education.”

II. VISION

Achieve excellence in Computer Applications with respect to teaching, learning and research to meet the growing needs of the industry and society.

III. MISSION

- Promote outcome-based learning strategies in-order to meet global industry standards.
- Encourage innovations and problem-solving capabilities in students and faculty.
- Cultivate collaborative research in both, students and faculty members through industry interactions and collaborations.
- Enhance entrepreneurship skills among students.

IV. PROGRAMME UNIQUE FEATURES

Keeping the view of National Education Policy, MCA Programme is designed with following features

- Students can choose any of the elective group through which he/she will be trained in specialized area for better career.
- Internship project provides a platform which gives acquaintance for solving IT problems.

V. PROGRAMME OBJECTIVES

- 1:** To build a strong foundation for students to become proficient in all academic concepts and technical skills necessary to become an IT Professional.
- 2:** To provide a conducive environment for designing, implementing and testing various software applications through Software Development.
- 3:** To keep the students and faculty abreast with the emerging technologies in the field of computer applications.
- 4:** To bring professionalism amongst the students and promote holistic development.
- 5:** To involve students in sustainable IT practices and community services.

VI. PROGRAMME OUTCOMES (PO)

PO1: Computational Knowledge: Apply knowledge of computing fundamentals, mathematics and given domain to design appropriate models for a given problem and/or requirements.

PO2: Problem Analysis: Apply fundamental knowledge of software engineering and various systems domain in order to analyze, identify, formulate and provide the solution to given problem.

PO3: Design/Development of Solutions: Design and evaluate solutions, systems, modules and processes for specified set of needs with appropriate consideration of societal values and industry expectations.

PO4: Conduct research in Computing problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Use of modern tools for delivering milestones like problem analysis, design, development, testing and deployment.

PO6: Professional Ethics: Learn and inculcate professional ethics, cyber regulations, professional responsibilities and norms of professional computing world.

PO7: Lifelong Learning: Acknowledge the need for continuous professional development and practice it through self-motivated, independent learning.

PO8: Management Domain: Involving in projects development as individual or group to solve problems in various domains and environments using computational and management skills.

PO9: Communication Efficacy: Demonstrate efficacy in verbal and non-verbal means of communication like reports, design documentation and presentations to elaborate about complex computing.

PO10: Innovation and Entrepreneurship: Provide conducive environment for innovation and entrepreneurship leading to solutions for betterment of society.

VII. PROGRAMME SPECIFIC OUTCOMES

PSO1: Ability to learn the various programming languages with database concepts and development environment

PSO2 : Ability to apply theoretical and practical knowledge to solve business problems in effective software solution through data communication technology concepts.

PSO3 : Enrich the knowledge in the areas of Advanced technologies and business practices.

PSO4 : Foster analytical and critical thinking abilities for efficient programming

PSO5 : Flourish the innovation and research attitude to develop IT artefact.

PSO6 : Maintain the personality with environmental and social concerns

VIII. ELIGIBILITY CRITERIA :

Admission to the programme is open to any Graduate (10+2+3) of any recognized University satisfying the following conditions.

1. Passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree. OR Passed B.Sc./ B.Com./ B.A. with additional bridge Courses (Bridge Course I/ Bridge Course II)as per the norms of the University.

OR

Passed any graduation degree (e.g. BE/ BTech/ BSc/BCom/BA/B.Voc/ etc) preferably with mathematics at 10+2 level or at Graduation Level

2. The candidate should have secured at least 50% marks (45% for SC/ST) in aggregate at graduate level university examination.
3. For students having no Mathematics background compulsory bridge course framed by the Bharati Vidyapeeth (Deemed to be University) and additional bridge course related to computer subjects should be completed
4. The candidate studying in final year of Bachelor's degree may also apply. Admission of such candidates will remain provisional until submission of final result certificates in original.
5. Subject to the above conditions, the final admission of final admission is based solely on –
 - a. The merit at All India Entrance Test conducted by Bharati Vidyapeeth (Deemed to be University), Pune.
 - b. Submission of Migration Certificate, Transference Certificate, anti-ragging affidavit etc.

IX. DURATION OF THE PROGRAMME

The duration of this programme is two years divided into four semesters or a minimum of 95 credits whichever is later.

The medium of instruction and examination will be only English.

X. MOOC Policy :

The Bharati Vidyapeeth (Deemed to be University), Pune offering MOOCS stands for Massive Open Online Courses Subjects. The student will complete MOOC courses prescribed by Institute from following sources in respective semester and will be evaluated based on the scores obtained by the Student/Learner in MOOCs.

Following are the sources from where Students/Learners can undertake MOOCs

1. iimb.ac.in
2. swayam.gov.in
3. edx.org
4. nptel.com (technical courses)
5. Coursera
6. National Programme on Technology Enhanced Learning (NPTEL)

Important Note:

- Students can complete MOOCs anytime during the valid tenure of the Programme
- Students have to submit completion Certificate of MOOCs. Unless certificate of all 03 MOOCs submitted, Fourth Semester Marksheet will not be issued.

XI. SCHEME OF EXAMINATION:

For some courses there is Internal Assessment (IA) conducted by the respective institutes continuously as well as a University Examination (UE) at the End-of-the Term. UE will be conducted out of 60 marks and IA will be conducted for 40 marks then these are converted to grade points and grades as per the Table I. For courses having only Continuous Assessment (CA) the respective institutes will evaluate the students in varieties of ways during the term for a total of specified marks. Then the marks will be converted to grade points and grades using the Table I.

XII. STANDARD OF PASSING:

For all courses, both UE and IA constitute separate heads of passing (HoP). In order to pass in such courses and to earn the assigned credits, the student/learner must obtain a minimum grade point of 5.0 (40% marks) at UE and also a minimum grade point of 5.0 (40% marks) at IA.

If Student fails in IA, the learner passes in the course provided, he/she obtains a minimum 25% marks in IA and GPA for the course is at least 6.0 (50% in aggregate). The GPA for a course will be calculated only if the learner passes at UE.

A student who fails at UE in a course has to reappear only at UE as backlog candidate and clear the Head of Passing. Similarly, a student who fails in a course at IA he has to reappear only at IA as backlog candidate and clear the Head of Passing to secure the GPA required for passing. The 10 point Grades and Grade Points assign according to the following table

Range of Marks (%)	Grade	Grade Point
$80 \leq \text{Marks} \leq 100$	O	10
$70 \leq \text{Marks} \leq 80$	A+	9
$60 \leq \text{Marks} \leq 70$	A	8
$55 \leq \text{Marks} \leq 60$	B+	7
$50 \leq \text{Marks} \leq 55$	B	6
$40 \leq \text{Marks} \leq 50$	C	5
Marks < 40	D	0

Table I : Grade Points and Grades

The performance at UE and IA will be combined to obtain GPA (Grade Point Average) for the course. The weights for performance at UE and IA shall be 60% and 40% respectively.

GPA is calculated by adding the UE marks out of 60 and IA marks out of 40. The total marks out of 100 are converted to grade point, which will be the GPA.

Rules of ATKT

For course up to four semesters, a student is allowed to carry any number of Backlogs of a prescribed course in Sem-I, II, III to Sem-IV provided he appears and have backlogs.

A student can appear for any four continuous semesters in an examination season including the regular semester, provided the student has appeared and have backlogs for other three semesters.

Formula to calculate Grade Points (GP)

Suppose that “Max” is the maximum marks assigned for an examination or evaluation, based on which GP will be computed. In order to determine the GP, Set $x = \text{Max}/10$ (since we have adopted 10 point system). Then GP is calculated by the following formulas

Range of Marks	Formula for the Grade Point
$8x \leq \text{Marks} \leq 10x$	10
$5.5x \leq \text{Marks} \leq 8x$	Truncate (M/x) +2
$4x \leq \text{Marks} \leq 5.5x$	Truncate (M/x) +1

Two kinds of performance indicators, namely the Semester Grade Point Average (SGPA) and the Cumulative Grade Point Average (CGPA) shall be computed at the end of each term. The SGPA measures the cumulative performance of a learner in all the courses in a particular semester, while the CGPA measures the cumulative performance in all the courses since his/her enrollment. The CGPA of learner when he/she completes the programme is the final result of the learner.

The SGPA is calculated by the formula

$$SGPA = \frac{\sum C_k * GP_k}{\sum C_k}$$

where, C_k is the Credit value assigned to a course and GP_k is the GPA obtained by the learner in the course. In the above, the sum is taken over all the courses that the learner has undertaken for the study during the Semester, including those in which he/she might have failed or those for which he/she remained absent.

The SGPA shall be calculated up to two decimal place accuracy.

The CGPA is calculated by the following formula

$$CGPA = \frac{\sum C_k * GP_k}{\sum C_k}$$

where, C_k is the Credit value assigned to a course and GP_k is the GPA obtained by the learner in the course. In the above, the sum is taken over all the courses that the learner has undertaken for the study from the time of his/her enrollment and also during the semester for which CGPA is calculated. The CGPA shall be calculated up to two decimal place accuracy.

The formula to compute equivalent percentage marks for specified CGPA:

% marks (CGPA)	1C* CGPA-10	If 5.00 ≤ CGPA ≤ 6.00
	5* CGPA+20	.00If 6.00 ≤ CGPA ≤ 8.00
	1C* CGPA-20	f If 8.00 ≤ CGPA ≤ 9.00 8.C 0
	2C* CGPA-110	f If 9.00 ≤ CGPA ≤ 9.50 9.C 0
	4C* CGPA-300	f If 9.50 ≤ CGPA ≤ 10.00 9.5 0

**XIII.
Award
of
Honors
:**

A

student who has completed the minimum credits specified for the programme shall be declared to have passed in the programme. The final result will be in terms of letter grade only and is based on the CGPA of all courses studied and passed. The criteria for the award of honours are given below.

Range of CGPA	Final Grade	Performance Descriptor	Equivalent Range of Marks (%)
9.5≤CGPA ≤10	O	Outstanding	80≤Marks≤100
9.0≤CGPA ≤9.49	A+	Excellent	70≤Marks≤80
8.0≤CGPA ≤8.99	A	Very Good	60≤Marks≤70
7.0≤CGPA ≤7.99	B+	Good	55≤Marks≤60
6.0≤CGPA ≤6.99	B	Average	50≤Marks≤55
5.0≤CGPA ≤5.99	C	Satisfactory	40≤Marks≤50
CGPA below 5.0	F	Fail	Marks below 40

Important Note:

- Student or Learner is expected to write Two Research Papers and publish it in Peer Reviewed Journals.
- A Student /Lerner can carry any number of backlog paper till Semester-IV provided his/her academic term(s) is/are granted

XIII. Question Paper Patterns for University Examination:

Title of the Course

Day:

Total Marks: 60

Date:

Time: 3 Hours

Instructions:

a. Attempt any FOUR questions from Section I Each question carries 10 Marks.

b. Attempt any ONE questions from Section II Each question carries 20 Marks.

SECTION – I

It should contain 06 questions covering the syllabus & should test the conceptual knowledge of the students

Question	Marks
Q.1	(10 marks)
Q.2	(10 marks)
Q.3	(10 marks)
Q.4	(10 marks)
Q.5	(10 marks)
Q.6 Write Short Notes on ANY TWO	(10 marks)

SECTION – II

It should contain 02 questions covering the entire syllabus & should be based on application of the Concepts

Q.7.....	(20 marks)
Q.8.....	(20 marks)

Syllabus Structure of Programme (Total Credits: 95)

SEMESTER I

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
101	Object Oriented Programming using Java	Core Courses (Theory)	3	2	1	-	40	60	100
102	Applied Database Management Systems (ORACLE)	Core Courses (Theory)	4	2	1	2	40	60	100
103	Computer Organization and Architecture	Core Courses (Theory)	3	2	1	-	40	60	100
104	Lab on Object Oriented Programming Using Java	Core Courses (Practical)	2	-	-	4	40	60	100

105	Client-Side Web Development	Core Courses (Practical)	3	2	-	2	40	60	100
106	Python Programming	Core Courses (Practical)	2	1	-	2	50		50
107	Soft Skills for Computing Professionals	Skill Enhanceme nt	2	2	-	-	50	-	50
108	Organization Behaviour	Multi- disciplinary	2	2	-	-	50	-	50
109	Open Course - Any one from Bucket	Skill Enhanceme nt	2	2	-	-	50	-	50
Total			23	15	03	05	400	300	700

SEMESTER II

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
201	Data Structures using Python	Core Courses (Theory)	3	2	1	-	40	60	100
202	Object-Oriented Software Engineering & Agile Practices	Core Courses (Theory)	3	2	1	-	40	60	100
203	Operating Systems for Programmers	Core Courses (Theory)	3	2	1	-	40	60	100
204	Lab on Data Structures using Python	Core Courses (Practical)	3	-	-	6	40	60	100
205	Server-Side Web Development	Core Courses (Practical)	3	2	-	2	40	60	100
206	EG1-1	Elective Courses	3	2	1	-	100	-	100
207	EG2-1	Elective Courses	3	2	1	-	100	-	100
208	Minor Project - 1 *	Minor Project	3	-	-	6	40	60	50
209	Open Course	Skill Enhancement	2	2	-	-	50	-	50
Total			26	14	5	7	490	360	850

SEMESTER III

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
301	Digital Transformation & Emerging Technology	Core Courses (Theory)	3	2	1	-	40	60	100
302	Software Design Principles and Design Patterns	Core Courses (Theory)	3	2	1	-	40	60	100
303	Computer Networking & cloud computing	Core Courses (Theory)	3	2	1	-	40	60	100
304	Web Application Development using .NET	Core Courses (Practical)	3	2	-	2	40	60	100
305	Generative AI & Prompt Engineering	Core Courses (Practical)	3	2	-	2	40	60	100
306	EG1-2	Elective Courses	3	2	1	-	100	-	100
307	EG2-2	Elective Courses	3	2	1	-	100	-	100
308	Minor Project – 2*	Minor Project	3	-	-	6	-	100	50
309	Open Course - Any one from bucket of Open courses	Multi-disciplinary / Skill Enhancement	2	2	-	-	50	-	50
Total			26	16	05	05	450	400	850

* 208-Minor Project-1 and 308- Minor Project-2 courses may be completed at home practical 6 hours per week

SEMESTER IV

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
401	Seminar on Recent Trends in IT	Core Courses (Theory)	4	-	4	--	40	60	100
402	MOOC-1	MOOCs	2	-	2	-	-		
403	MOOC-2	Internship / Major Project	2	-	2	-	-		
404	Internship / Major Project		12			24		100	100
Total			20	-	8	12	40	160	200

Open courses Bucket

1	Cyber Security
2	Financial Literacy
3	IKS-Yoga and Wellness
4	Indian Constitution
5	Entrepreneurship Development & Start-up
6	Business Intelligence using (Tablelu & PowerBI)
7	AI Essentials
8	Basics of Data Analytics
9	Entrepreneurship Development and Start-up management
10	Soft Skills
11	Universal Human Values
12	Digital Technology
13	Human Psychology at Workplace
14	Water Management
15	Social Change in Technology
16	Economics for IT Industry

List of Elective Groups for 2 Year Programme :

Each student will opt for one elective group in 2nd year (EG1 and EG2)

Digital Commerce:

E1: Digital Business Models and E-Commerce

E2: Customer Analytics and Digital Marketing

Big Data:

E1: Introduction to Big Data,

E2: Big Data Analytics

AI-ML

E1: AI Foundations and Intelligent Systems

E2: Applied Machine Learning

Data Science:

E1: Data Analytics and Visualization

E2: Statistical Learning and Predictive Analytics

Cloud Computing

E1: Fundamentals of Cloud Computing

E2: Cloud Platforms and Application Deployment

IOT:

E1: Internet of Things Fundamentals and Applications

E2: Device Programming and Application Development

Full Stack Development

E1: API Design and Full-Stack Engineering

E2: Microservices and Cloud-Native Applications

Cyber Security

E1: Cyber Security and Data Protection

E2: Security Auditing and Risk Management

Mobile Computing

E1: Foundations of Mobile Computing

E2: Mobile Application Development

Blockchain Technology:

E1: Fundamentals of Blockchain Technology,

E2: Blockchain Technology and Applications\

XVII. Bridge Course:

(offered in Semester-1 to complete the eligibility criteria - Institutional Continuous Assessment)

- Mathematical Foundation (for students without mathematics background)
- Introduction to Programming (for students without computer science background)

Bridge Course I :

This course is designed and compulsory for the students from Non-Mathematics background and who have not completed mathematics in their 12th or graduation course. The course can be conducted concurrently with semester I courses. The evaluation of this course will be at institute level for 100 marks. The student must score minimum 40 marks to pass this course. There will be no credits assigned to this Bridge Course.

Subject Name	Bridge Course I Mathematical Foundation
No. of Credits	00
Course Objective	To prepare background of the student to study courses in MCA
Cognitive Abilities	Course Outcome as per Blooms Taxonomy
Remembering	Remembering basic concepts and their representations
Understanding	Understanding applications of various discrete structures like sets, relations, graphs etc.
Applying	Applying various structures to represent problem data.
Analyzing	Learn to analyze the data for the given problem for representing it using proper structure.
Evaluating	Evaluate the problem for proper discrete structures.
Creating	Design new structures based on basic discrete structures to represent data..
Text Books	Discrete Structures by Kenneth Rosen
Course Plan	
Unit	Content
1.	Set Theory : Definition of a set, Representation of elements of sets, Methods of representing sets , types of sets, operations on sets , cardinality of a set, Principle of Inclusion and Exclusion , Venn Diagram , Proof by using Venn diagram
2.	Functions and Relations : Definition of Function, Types of Functions ,Composite Function, Relation definition, representation of relations
3.	Logic: Propositions, Logic Operations-Negation, Disjunction, Conjunction, Conditional and Biconditional, Truth Tables of compound propositions, Translating English sentences in to logical statements and vice versa, Logic gates and circuits
4.	Matrices: Matrix Definition, General Form, Representation of matrix in computers, Types of matrices, Operations on matrices: Addition, Subtraction and Multiplication, transpose , row / column transformations , Inverse of the matrix by Co-factor and Adjoint method, solutions to three variable problems by using matrices, application problems of matrices
5.	Graphs - Graph terminologies, types of graphs , representation of graph in computers, Paths, Euler and Hamilton graphs, graph colorings.

Bridge Course II:

This course is designed and compulsory for the students from Non-IT background. The course can be conducted concurrently with semester I courses. The evaluation of this course will be at institute level for 100 marks. The student must score minimum 40 marks to pass this course. There will be no credits assigned to this Bridge Course.

Subject Name	Bridge Course II Introduction to Programming
No. of Credits	00
Pre Requisite	Basic Mathematics and MSCIT course
Cognitive Abilities	Course Outcome as per Blooms Taxonomy
Remembering	Basic formula for finding areas, volumes, graphical representation of data is to be remembered.
Understanding	Do calculations by using formulas, algorithm, C program structure are to be understood
Applying	Apply basic knowledge of mathematics and computers to write programming codes.
Analyzing	Analyze the problem to represent in proper format such as graphs, trees for effective working
Evaluating	Evaluate the programs or problems for algorithms, logic
Creating	Creating proper program logic so as to reduce lines of codes is expected
Unit	Content
1.	Algorithm ,flow charts, integers, division, relations, relations and their types, representation of relation in computer memory, number conversion systems
2.	Trees, applications of trees, tree traversal algorithms, minimum spanning trees
3.	Fundamentals of C programming, Keywords and Identifiers, Constants, Variables, Data types, Declaration of variables, Declaration of variables as constant, Operators, Types of operators, Input and Output functions - printf(), scanf(), getchar(), putchar(), Formatted input and formatted output.
4.	Control Statements- Sequential, Selection, Iteration Statements, Branching structure- if statement, if-else statement, Nested if-else statement, else if Ladder, Conditional operator, switch statement, Loop control structures- while loop, do-while loop, for loop, Nested for loop, Jump statements-break, continue, goto statements
5.	Function call, return statement, Function parameters, Types of functions, Arrays and functions
6.	Introduction to OOP concepts.
Text Books	1.Discrete Structures by Kenneth Rosen 2.C programming by Yashwant Kanetkar 1.Object Oriented Programming by Balguruswamy
Reference Books	C Programming language by Brain W. Kernighan

Detail Syllabus

Sem-I

Programme:MCA CBCS–Revised Syllabus w.e.f.-Year -2026-27			
Semester	Course Code	Course Title	
I	101	Object Oriented Programming using Java	
	Prepared By	Dr. Alok Shah	
Type	Credits	Evaluation	Marks
DSC	3	UE:IE	60:40:00

Unit No.	Contents	Sessions (Hrs.)	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
1	Introduction to Java Java Basics: Features of Java, History of Java, Installations of JDK and eclipse as IDE Writing and executing first Java program. Understanding role Java compiler, JVM, Understanding how Java is platform independent and secure. Java data types, variables, operators, expressions, type conversion and casting in Java. Control structures in java: if, if-else and switch statements, using iterative/looping statements in Java: while, do-while and for. Writing functions: Need of functions/methods, Writing and using static method; concepts of passing and returning values.	5		Lecture with PPTs ,Practicing programming problems	Understand, Apply, Analyse	Quiz, writing short answers
2	Class and Object Concepts: Introduction to Object Oriented concepts, Defining a class, creating objects from class, adding attributes and methods to the class, using constructors, Java naming conventions for class, properties and methods/functions. Passing values to the functions – pass by value, pass by reference, Function overloading. Modifiers – public, private,	5		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers

	protected, default, static, final Understanding use of Wrapper classes and Garbage collection in Java					
3	Arrays and Strings One dimensional arrays, Multidimensional arrays, exploring String class and methods, String Buffer class. Packages - creating and accessing a package, importing, packages, creating user defined packages, Concept of package. Introduction to Exception Handling and user defined exceptions.	6		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers
4	Inheritance and Polymorphism : Concept and importance of inheritance, is-a relationship, types of inheritance, Polymorphism– function overriding, dynamic method dispatch. Overriding methods with throws clause. Using abstract and final keywords with class declaration, Concept of interface, Comparison of Interface and class. Access modifiers and data accessibility in derived classes, method access modifier and method overriding.	5		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers
5	Concurrent Programming: Concept of threads, lifecycle of threads, creating threads, Thread class, Runnable interface, Thread synchronization, inter thread communication – wait(), notify(), notify All() methods.	6		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers
6	Java Input/Output : Concept of streams, types of streams– bytestreams, characterstreams, The Console: System.out, System.in, and System.err Understanding Fileclass, InputStream class, OutputStream class, FileInputStreams, FileOutputStream, Using character oriented Reader and Writer class, FileReader, FileWriter. Introduction to Buffered streams – DataInput and DataOutput Streams using BufferedReader, BufferedWriter. Making use of ObjectStreams for Serialization and deserialization	6		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers

7	Java Collections and Utility Classes: Introductions to generics: generic types and methods Collection Basics-A Collection Hierarchy, Using ArrayList and Vector, LinkedList, making use of Iterator to access collection elements.Set:HashSet,LinkedHashSet , TreeSet , Role of Comparable and Comparator interfaces, Introduction to Map: Hashmap,HashTable, TreeMap, LinkedHashMap.	5		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers
8	Applet and GUI Programming AWT Components,Swing Components,Event Handling,Layout Managers Building Simple GUI Applications.Applet Life Cycle,Drawing Different Shapes on Applet,GUI Application Using Applet.	7		PPTs, Practicing programming problems	Understand, Analyse, Apply	Quiz, writing short answers

Sr.No.	Name of the Author	Title of the Book	Year Edition	Publisher Company
1	Herbert Schildt	Java: The Complete Reference	Seventh Edition, 2007	McGraw-Hill Osborne Media
2	Cay S. Horstmann and Gary Cornell	Core Java-Volume-I	Eighth Edition, 2008	Sun Core Series
3	Bruce Eckel	Thinking In Java	Fourth Edition	Printice Hall

MOOCS

Resources No.	Website address
---------------	-----------------

1	NPTEL
---	-------

Semester	Course Code	Course Title	
I	102	Applied Database Management Systems	
	Prepared by		
Type of Course	Credits	Evaluation	Marks
DSC	4	UE:IE	60:40

Course Objectives:

- To teach the fundamentals of database systems at a master level. A variety of topics will be covered that are important for modern databases in order to prepare the students for real life applications of databases.
- To impart knowledge of the concepts related to databases and operations on databases. It also gives the idea how database is managed in various environments with emphasis on security measures as implemented in database management systems.

After completing the course the students shall be able to-

CO1: Remember the database concepts

CO2: Understand the concept of database and techniques for its management

CO3: Understand data security standards and methods.

CO4: Understand the fundamentals of Distributed Database Systems

CO5: Design different data models at conceptual and logical level and translate ER Diagrams to Relational Data Model.

CO6: Normalize the database.

CO7: Identify and study the file organization schemes for DBMS.

CO8: State and Describe features for Concurrency and Recovery.

CO9: Convert the relational algebra statements to the SQL statements.

CO10: Design the queries using Relational Algebra.

Unit	Content	Sessions	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
1	Difference between Data, Information, Data Processing & Data Management. File Oriented Approach, Database oriented approach to Data Management, Need for DBMS, Characteristic of Database, Database Architecture: Levels of Abstraction, Database schema and instances, 3 tier architecture of DBMS, Data Independence. Database users, Types of Database System. Database Languages, DBMS interfaces.	08	CO1 CO2	Lecture with PPTs, Discussion	Understand	Discussion

2	Data Modelling in Database Data Models, Logical Data Modelling: Hierarchical Data Model, Network Data Model, Relational Data Model. Conceptual Data Modelling: Entity Relationship Model, Entities, Attributes, Types of Attributes, Relationships, Relationship set, Degree of relationship Set, Mapping Cardinalities, Keys, ER Diagram Notations, Roles Participation: Total and Partial, Strong and Weak Entity Set. The extended entity relationship (EER) model, Subclass, Superclass, generalization, specialization, Attribute Inheritance. Relational Data Model: Codd's Rules for RDBMS, Translating ER Diagram to Relational Database.	09	CO5	Lecture with PPTs, Practical sessions on computer	Understand the Models and analyse	Understand and draw the models of database
3	Normalization and Relational Algebra Normalization Vs De- Normalization, Decomposition, Lossy and Lossless Decomposition, Functional Dependencies, Normal forms 1NF, 2NF, 3NF, BCNF, Case Studies on Normalization. Relational Algebra: Keys: Composite, Candidate, Primary, Secondary, Foreign, Relational Algebra Operators: Select, Project, Divide, Rename. Set Operations: Union, Intersect, Difference, And Product, Joins: Outer Joins, Inner Joins with example.	09	CO6 CO9 CO10	Lecture with PPTs, Case Studies	Understand and analyze	Analyze and practice the case studies on various
4	File Structures and Data Administration File Organization, Overview of Physical Storage Media, Magnetic Disk, RAID, Tertiary Storage, Storage Access, Data Dictionary Storage, Organization of File (Sequential, Clustering), Indexing and Hashing, Basic Concepts, indices, B+ Tree index file, B- tree index file, Static hashing, Dynamic Hashing	08	CO7	Lectures with PPTs,	Evaluate	Formulate and practice the case studies on various topics
5	Concurrency Control And Recovery Techniques Concurrency Control: Single User and Multiuser systems, Multiprogramming and Multiprocessing, Basic Database access operations, Concept of transaction, transaction state, ACID properties, Schedules, Serializability of schedules., Concurrency Control, Need for Concurrency control, lock based protocols, timestamp based protocols, Multiple granularity, Multiple Version Techniques, Deadlock and its handling, Wait-Die and Wound-Wait, Deadlock prevention without using timestamps, Deadlock detection and time outs, Starvation Recovery Techniques:	09	CO8	Lectures with PPTs,	Compose and execute	Discussion

	Database Recovery, Types of Failures, Storage Structure: Volatile, Non Volatile and stable storage, Data access. Recovery and atomicity, Recovery Techniques / Algorithms: Log Based Recovery, Check points, Shadow Paging.					
6	Data Administration And Security Data administration, Role and Responsibility of DBA, Creating/Deleting/Updating table space, Database Monitoring, User Management. Basic data security principles – user privileges, data masking, encryption and decryption. Data Security Implementation, revalidation of user, role, privileges. Data Quality Management, Basic quality principles, data quality audit, data quality improvement	09	CO3	Lectures with PPTs	Demonstrate	Discussion
7	Introduction to Distributed Database, NOSQL and Mongo DB Heterogeneous and Homogeneous Databases, Distributed database features and needs, Advantages and Disadvantages, Distributed Database Architecture. Levels of distribution, transparency and replication. Fragmentation. PL/SQL – Cursor, Stored Procedure and Functions, Packages, Triggers. • Introduction to NoSQL – Architecture, Sharding, Replica sets. • NoSQL Assumptions and the CAP Theorem • Strengths and weaknesses of No SQL • MongoDB Functionality Examples	08	CO4	Lectures with PPTs, Write NoSQL and Mongodbs Documents	Compose and execute	Understand and calculate cost of project

Reference Books:

Sr.No.	Name of the Author	Title of the Book	Year Edition	Publisher Company
1	Ramez Elmasri, Shamkant B. Navathe	Fundamentals of Database Systems	5 th Edition	PEARSON Education
2	ASilberschatz, H Korth, S Sudarshan	Database System and Concepts	4 th Edition	McGraw-Hill.
3	Shakuntala Gupta Edward	Practical MongoDB	--	Navin Sabharwal published by APress

Online Resources:

Online Resources No.	Website address
1	https://www.w3schools.com/sql/
2	https://www.tutorialspoint.com/sql/index.htm 3
3	https://www.javatpoint.com/sql-tutorial

MOOCs:

Resources No.	Website address
1	https://www.coursera.org/learn/intro-sql
2	https://www.coursera.org/projects/introduction-to-relational-database-and-sql
3	https://www.coursera.org/projects/intermediate-rdb-sql

Semester	Course Code	Course Title				
I	103	Computer Organization and Architecture (COA)				
	Prepared by	Prof. Dr. Kamal Alaskar				
Type of Course	Credits	Evaluation	Marks			
CC	4	UE	60			
Course Objectives:						
1. Understand basic computer organization and architecture. 2. Learn data representation and arithmetic operations. 3. Study CPU design, instruction sets, and memory systems. 4. Explore I/O organization and advanced processing techniques. 5. Introduce IoT architecture, communication protocols, and embedded computing concepts.						
Course Outcomes (COs):						
CO1 – Explain the basic structure and functional units of a computer system. CO2 – Apply number systems and data representation methods. CO3 – Analyze instruction set architectures and addressing modes. CO4 – Analyze CPU organization and control mechanisms. CO5 – Explain memory organization and hierarchy. CO6 – Apply I/O organization, interrupts and DMA concepts. CO7 – Evaluate pipelined and parallel processing systems. CO8 – Explain IoT architecture, devices and communication protocols.						
Unit	Content	Sessions	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
1	Basic Structure of Computer Systems: Functional units, Von Neumann architecture, instruction cycle, bus structures.	5	CO1	Lectures, PPT	Understand	Assignment
2	Data Representation & Arithmetic: Number systems, complements, IEEE-754, arithmetic operations.	5	CO2	Problem Solving	Apply	Class Test
3	Instruction Set Architecture (ISA): formats, addressing modes, RISC vs CISC, assembly concepts.	7	CO3	Lecture, Programming	Analyze	Mid Term
4	CPU Organization & Design: Register organization, ALU, control unit, micro-operations.	7	CO4	Diagram Explanation	Analyze/Evaluate	Assignment
5	Memory Organization: hierarchy, cache, virtual memory, secondary storage.	5	CO5	Demonstration	Analyze	Class Test
6	Input/Output Organization: I/O interface, interrupts, DMA, priority handling.	5	CO6	Discussion	Apply/Analyze	Assignment
7	Pipeline & Parallel Processing: hazards, speedup, multiprocessors, throughput.	6	CO7	Presentation	Evaluate/Create	End Term Exam
8	Internet of Things (IoT): IoT architecture, sensors and actuators, embedded systems, MQTT, CoAP, edge/cloud	5	CO8	Lecture, Case Study	Understand/Apply	Quiz, Assignment

	computing, applications and security.					
--	---------------------------------------	--	--	--	--	--

Reference Books :

Sr.No.	Author	Title of the Book	Edition	Publisher
1	William Stallings	Computer Organization and Architecture: Designing for Performance	11th	Pearson
2	M. Morris Mano	Computer System Architecture	3rd	Pearson
3	Carl Hamacher et al.	Computer Organization and Embedded Systems	6th	McGraw Hill
4	Patterson & Hennessy	Computer Organization and Design (RISC-V Edition)	2nd	Morgan Kaufmann
5	Andrew S. Tanenbaum & Todd Austin	Structured Computer Organization	6th	Pearson
6	Arshdeep Bahga & Vijay Madisetti	Internet of Things: A Hands-On Approach	Latest	Universities Press
7	Raj Kamal	Internet of Things: Architecture and Design Principles	Latest	McGraw Hill
Online Resources No.		Website address		
1		https://nptel.ac.in		
2		https://www.coursera.org		
3		https://www.edx.org		

Programme:MCA CBCS–Revised Syllabus w.e.f.-Year -2026-27			
Semester	Course Code	Course Title	
I	104	Lab on Object Oriented Programming using Java	
	Prepared By	Dr. Alok Shah	
Type	Credits	Evaluation	Marks
DSC	2	UE:IE	60:40:00

Unit	Contents	Sessions (Hrs)	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
Introduction to Java	Display 'Welcome to Java Programming'	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Print personal details					
	Demonstrate primitive data types					
	Arithmetic operations on two numbers					
	Type casting and type conversion					
Control Statements	Largest of three numbers	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Even or odd					
	Leap year					
	Multiplication table					
	Factorial					
	Fibonacci series					
	Prime number check					
	Palindrome number					
Methods and Functions	Method to calculate square	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Method to find maximum					
	Method overloading					
	Passing parameters and returning values					
	Recursive factorial					
Classes and Objects	Student class	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Employee class					
	Bank Account class					
	Rectangle class					
	Book class					

Constructors and Modifiers	Default and parameterized constructors	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Constructor overloading					
	this keyword					
	static variables and methods					
	final keyword					
Wrapper Classes and Garbage Collection	Primitive-wrapper conversions	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Auto boxing and unboxing					
	Garbage collection					
Arrays	Sum and average of array	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Largest and smallest element					
	Sort array					
	Linear search					
	Matrix addition					
	Matrix multiplication					
	Transpose of matrix					
Strings	Count vowels and consonants	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Reverse a string					
	Palindrome string					
	Count words in sentence					
	String methods					
	StringBuffer methods					
Packages	Create user-defined package	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Import and use package					
	Package for mathematical operations					
Exception Handling	Divide-by-zero exception	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	ArrayIndexOutOfBoundsException					
	NumberFormatException					
	Multiple catch blocks					
	throw and throws					
	User-defined exception					
Inheritance	Single inheritance	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Multilevel inheritance					
	Hierarchical inheritance					
	super keyword					

	Constructor invocation in inheritance				ze	
Polymorphism	Method overriding	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Dynamic method dispatch					
	Abstract class implementation					
	Final class and method					
Interfaces	Create and implement interface	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Multiple interfaces					
	Interface inheritance					
	Abstract class vs interface					
Multithreading	Thread class	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Runnable interface					
	Thread priorities					
	Thread synchronization					
	wait(), notify(), notifyAll()					
File Handling	Read from file	2		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Write to file					
	Copy file contents					
	Count words, characters and lines					
	FileReader and FileWriter					
	BufferedReader and BufferedWriter					
Serialization	Serialize Student object	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	Deserialize Student object					
	Store multiple objects					
Collections Framework	ArrayList	1		Lab Demo	Understand, Apply, Analyze	Quiz, testing programming skills through practical test.
	LinkedList					
	Vector					
	HashSet					
	LinkedHashSet					
	TreeSet					
	HashMap					
	TreeMap					
	Iterator					
	Comparable and Comparator					
Generics	Generic class	1		Lab	Understa	Quiz, testing

	Generic method			Demo	nd, Apply , Analy ze	programming skills through practical test.
	Bounded type parameters					
GUI Program ming using AWT	Simple AWT form	2		Lab Demo	Understa nd, Apply , Analy ze	Quiz, testing programming skills through practical test.
	Label, TextField and Button					
	Checkbox and RadioButton					
	Event handling					
	Login form					
	Calculator					
	Student information form					
	Layout managers					
	Menu-driven application					
Applet Program ming	Simple applet	2		Lab Demo	Understa nd, Apply , Analy ze	Quiz, testing programming skills through practical test.
	Draw lines and rectangles					
	Draw circles and ovals					
	Draw different shapes and colors					
	Interactive applet					

Reference Books.

Sr.No.	Name of the Author	Title of the Book	Year Edition	Publisher Company
1	Herbert Schildt	Java: The Complete Reference	Seventh Edition, 2007	McGraw-Hill Osborne Media
2	Cay S. Horstmann and Gary Cornell	Core Java-Volume-I	Eighth Edition, 2008	Sun Core Series
3	Bruce Eckel	Thinking In Java	Fourth Edition	Printice Hall

MOOCS

Resources No.	Website address
1	NPTEL

Semester	Course Code	CourseTitle	
I	105	Client-Side Web Technologies (HTML JavaScript and CSS)	
	Prepared by	Dr. Suvarna Patil	
Type of Course	Credits	Evaluation	Marks
CC	3	UE(60)+IE(40)	100
Course Objectives:			
Objectives :			
To learn Web Supporting Technologies and develop website			
Course Outcomes:			
CO1: To remember basic concepts of Web Supporting Technologies. CO2: To understand syntaxes of HTML, HTML5, CSS and JavaScript CO3: To design web pages by applying HTML, HTML5, CSS and JavaScript. CO4: To analyze and solve real life problem using web supporting technology given in the syllabus.			

Unit	Content	Sessions (Hrs)	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
1	Overview of HTML , concept of Tag, types of HTML tags, structure of HTML program, Text Formatting Through HTML Emphasizing Material in a Web Page, Using Image tag, attributes of Image tag, Block Element, Inline Elements. Lists: Unordered, Ordered, lists. Tables: Add Table, Define header rows & data rows, use of caption tag, changing height & width of table, BGcolor, cell padding, cell spacing, colspan, row span, images in table, Frames: Introduction To frames, using frames & framesets, named frames. Hyperlink - Types of hyperlinks, linking to the beginning of document, linking to a particular location in a document, image as hyperlinks	7	CO1, CO2	Explanation, Demo, PPT	Understand	Q-A in class, Quiz, theory assignment, Lab assignments

2	HTML5 Introduction to HTML5, Features of HTML5, Elements of HTML5, HTML Media and Graphics	8	CO1, CO2	Explanation Demo, PPT,	Remember	Q-A in class, Quiz, theory assignment, Lab assignments
	Cascading Style Sheets Introducing CSS, Types of style sheets: inline, embedded and external style sheets, working with CSS properties: text properties, color and background properties, border and shading, box and block properties, positioning with CSS, various types of CSS selectors, Using class and span tag, External style sheets,	8	CO1, CO2, CO3	Explanation, Demo, PPT	design	Q-A in class, Quiz, theory assignment, Lab assignments ,Case based example solving
4	Introduction to JavaScript (Client-Side Scripting): Introduction to scripting, overview of Java Script, advantages, client-side java Script, capturing user input, writing JavaScript into HTML, Advantages and limitations of JavaScript, JavaScript Basics: Data types, literals, variables and operators, Java Script arrays, array methods, expressions. JavaScript Programming Constructs: Assignment, data declaration, if, switch, while, for, do while, label, break, continue, function call, return, with, delete, method of invocation Dialog boxes -Alert dialog box, prompt dialog box, confirm dialog box, window objects JavaScript Functions- Types of functions in Java Script- Built in functions, User defined functions, function declaration, passing parameters, variable scope, return values.	12	CO1, CO2, CO3, CO4,	Explanation, Demo, PPT	Analyze	Mini projects, team work,

5	Forms: Interactive web pages concepts, difference between static & dynamic web pages, Concept of form, how form works, Different elements - text, password, button, submit, reset, checkbox, Radio, Text Area, select & option, properties of form elements, form object's Method, Other built-in Object: String object, math object, date object, Regular Expressions, Form validation JavaScript Events: What is an Event? Onclick Event Type, onsubmit Event Type, onmouseover and onmouseout, onchange, onload, onkeydown, working with DOM.	10	CO1, CO2, CO3, CO4	Explanation, Demo, PPT	Create	Case Presentation Activity End Term: Theory Applied
---	---	----	--------------------	------------------------	--------	--

Reference Books

Sr.No.	NameoftheAuthor	TitleoftheBook	Year Edition	Publisher Company
1	Dromey	How to solve computer	2015,3 rd edition	PHI Publication
2	P. K. Sinha	Computer Fundamentals	12 th edition	PBP Publication
3	V. Rajaraman	Computer Fundamentals	6TH EDN. 2014	PHI Publication

Online Resources

OnlineResourcesNo.	Websiteaddress
1	www.edx.com
2	www.coursera.com

MOOCs:

ResourcesNo.	Websiteaddress
1	Alisons
2	Swayam

Semester	Course Code	Course Title	
1	106	Python Programming	
Type	Credits	Evaluation	Marks
Core Courses (practical)	2	IA	50
Course objectives			
- To introduce the fundamentals of Python programming, including syntax, data types, operators, control structures, and program execution, enabling students to develop basic Python programs. - To develop an understanding of object-oriented programming concepts such as encapsulation, inheritance, polymorphism, and abstraction, and implement these concepts using Python classes and objects. - To provide practical knowledge of file handling, exception handling, and Python standard library modules for developing reliable, maintainable, and real-world software applications. - To enhance students' analytical and problem-solving abilities by evaluating Python data structures and algorithm efficiency, enabling them to select appropriate computational solutions for different problem domains.			
Course Outcome			
CO1 - Recall and describe Python's data types, control structures, and execution model to write syntactically correct programs. (Remember) CO2 - Explain object-oriented principles — encapsulation, inheritance, polymorphism, and abstraction — and illustrate them using Python class hierarchies. (Understand) CO3 - Apply file I/O, exception handling, and standard library modules to build robust, real-world Python applications. (Apply) CO4 - Analyse data structures (lists, dicts, sets, tuples) and algorithm complexity to select optimal solutions for computational problems. (Analyse)			

Unit Name	Sub Unit	Competency	Competency Indicator	Ses sio ns
Unit I Python Foundations and Environment CO1 <i>Remember / Understand</i>	Python Overview, Installation & Environment Setup	Set up and navigate a Python development environment	Installs Python and configures an IDE (CPython/Anaconda/VS Code); runs a first script successfully	8
	Data Types, Variables & Type Casting	Recall and apply Python's core data types and typing behaviour	Correctly identifies and converts between int, float, complex, str, bool and NoneType in code	
	Operators & Control Flow	Construct decision-making and iterative logic	Writes correct if-elif-else and for/while constructs using appropriate operators and control statements	
	Functions,	Design	Defines functions using default args,	

	Arguments & Scope	reusable functions with correct scope handling	*args/**kwargs and lambda; explains and applies the LEGB rule	
	Recursion	Apply recursive thinking to problem solving	Writes and traces recursive functions for standard problems (e.g. factorial, Fibonacci)	
	Built-in Functions, Modules & Packages	Utilise Python's built-in functions and module system	Uses map, filter, zip, enumerate and sorted correctly; imports modules and installs packages via pip	
Unit II Data Structures and Collections <i>CO4 · Apply / Analyse</i>	Lists & Tuples	Apply core sequence data structures	Creates and manipulates lists (slicing, comprehensions, nesting) and tuples (packing/unpacking, named tuples)	8
	Dictionaries, Sets & Frozensets	Apply mapping and set-based data structures	Implements dict comprehensions, defaultdict/OrderedDict, and set operations for a given use case	
	Strings, Formatting & Regex	Apply string formatting and pattern-matching techniques	Formats strings using f-strings and extracts/matches patterns using the re module	
	Iterators & Generators	Apply lazy-evaluation constructs	Writes generator functions using yield and generator expressions to process data streams	
	Stacks, Queues & Deques	Apply the collections module to linear data structures	Implements stack/queue/deque operations using collections.deque for a given scenario	
	Complexity Analysis, Sorting & Searching	Analyse algorithmic efficiency and select optimal solutions	Determines Big-O complexity of code; implements and compares bubble/merge/quick sort and linear/binary search	
Unit III Object-Oriented Programming in Python <i>CO2 · Apply / Analyse</i>	Classes & Objects	Apply class-based design fundamentals	Defines classes using __init__ and self; differentiates instance variables from class variables in code	8
	Encapsulation	Apply access-control conventions	Implements public, protected (__) and private (__) attributes and justifies their appropriate use	
	Inheritance	Apply inheritance hierarchies	Builds single, multiple and multilevel inheritance; traces MRO and uses super() correctly	
	Polymorphism	Analyse	Demonstrates method overriding and	

	& Abstraction	polymorphic and abstract class designs	duck typing; implements abstract classes using ABCMeta/@abstractmethod	
	Magic Methods & Decorators	Apply Python's special methods and built-in decorators	Overrides <code>__str__</code> / <code>__repr__</code> / <code>__len__</code> / <code>__eq__</code> / <code>__lt__</code> ; applies <code>@staticmethod</code> , <code>@classmethod</code> and <code>@property</code> appropriately	
	Exception Handling & Context Managers	Apply robust error-handling and resource-management patterns	Writes try-except-else-finally blocks with custom exceptions; implements context managers using <code>__enter__</code> / <code>__exit__</code>	
Unit IV File Handling, Databases and APIs <i>CO3 · Apply / Evaluate</i> Total: 6 sessions	File I/O & Structured File Formats	Apply file-handling techniques to structured data formats	Reads and writes text and binary files; parses and generates CSV, JSON and XML data correctly	6
	SQLite & CRUD Operations	Apply database operations within Python applications	Connects to SQLite using the <code>sqlite3</code> module and performs Create, Read, Update and Delete operations	
	REST API Consumption & JSON Handling	Apply API integration techniques	Sends requests using the <code>requests</code> library and parses JSON responses into usable Python objects	
	Web Scraping	Evaluate ethical and technical approaches to web scraping	Extracts data using BeautifulSoup while applying ethical scraping guidelines (robots.txt, rate limiting)	

Recommended resources

Textbooks

Mark Lutz —**Learning Python**, 5th ed., O'Reilly Media (2013)

Eric Matthes —**Python Crash Course**, 3rd ed., No Starch Press (2023)

Luciano Ramalho —**Fluent Python**, 2nd ed., O'Reilly Media (2022)

Online / MOOCs

Coursera — Python for Everybody Specialisation by University of Michigan (coursera.org)

NPTEL — Programming, Data Structures and Algorithms using Python by IIT Madras (nptel.ac.in)

Real Python — Practical Python tutorials and projects (realpython.com)

Semester	Course Code	Course Title	
I	107	Soft Skills for Computing Professionals	
	Prepared By	Dr. Satyawan Hembade	
Type	Credits	Evaluation	Marks
Skill Enhancement	2	IE	50
Course Objectives:			
• Develop effective communication skills for professional interactions through oral, written, and digital communication in computing environments. • Enhance interpersonal and teamwork abilities to enable students to collaborate effectively in software development teams using professional work practices. • Cultivate critical thinking, problem-solving, and ethical decision-making required for addressing workplace challenges and maintaining professional integrity. • Build leadership, emotional intelligence, and adaptability to prepare students for dynamic and multicultural computing workplaces. • Prepare students for career success by developing employability skills, including resume writing, interview techniques, professional networking, and responsible use of AI productivity tools.			
Course Outcomes:			
CO1: Communicate effectively in technical and professional environments using oral, written, and digital communication. Apply (C3) CO2: Collaborate effectively in multidisciplinary software development teams by demonstrating teamwork, leadership, emotional intelligence, and conflict resolution skills. Apply (C3) CO3: Apply critical thinking, ethical reasoning, and professional conduct while solving workplace problems in computing environments. Analyze (C4) CO4: Demonstrate career readiness through resume preparation, interview skills, professional networking, and responsible use of AI tools. Create (C6)			

Unit	Contents	Sessions (Hrs)	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
1	Professional Communication for Computing • Communication process and barriers • Active listening and questioning techniques • Verbal and non-verbal communication • Professional email and chat etiquette • Technical writing ◦ Bug reports ◦ Meeting minutes ◦ Software documentation • Presentation skills for technical audiences • Cross-cultural communication in global software teams Activities • Elevator pitch • Technical presentation • Email writing exercises	8	CO1	Interactive lectures, demonstrations, role plays, presentation practice, peer feedback, technical writing exercises	Understand (C2), Apply (C3)	Presentation, email writing, technical documentation, oral communication assessment, quizzes
2	Teamwork and Workplace Effectiveness • Team dynamics in Agile environments • Collaboration and accountability • Emotional Intelligence (EQ) • Conflict management and negotiation • Time management and productivity • Critical thinking and creative problem solving • Adaptability and continuous learning Activities • Scrum simulation • Group discussion • Team problem-solving challenge • Reflection journal	8	CO2, CO3	Group discussions, Agile team simulation, case studies, collaborative learning, brainstorming, reflective learning	Apply (C3), Analyze (C4)	Group activity, case analysis, peer evaluation, teamwork assessment, reflection journal

3	Professionalism, Ethics and Career Readiness - Professional ethics for software professionals - Workplace etiquette and organizational behaviour - Digital citizenship and online professionalism - Responsible use of AI tools (prompting, verification, privacy, plagiarism) - Resume and LinkedIn profile preparation - Interview skills (HR and technical) - Personal branding and networking Activities - Mock interviews - Resume review - LinkedIn profile creation - AI-assisted communication exercises	8	CO1, CO2, CO2, CO4	Case studies, mock interviews, resume workshop, AI-assisted exercises, industry interaction	Analyze (C4), Evaluate (C5), Create (C6)	Resume evaluation, mock interview, ethical case analysis, LinkedIn profile assessment, AI prompt assignment
4	Capstone Professional Skills Workshop (6 Hours) Integrated practice through: - Group presentation - Software project pitch - Case studies - Role plays - Peer feedback - Reflection and self-assessment Suggested Pedagogy - Interactive lectures - Case studies - Group discussions - Role plays - Mock interviews - Peer assessment - Industry talks - Presentation sessions - AI-assisted writing and communication exercises	6	CO1, CO2, CO2, CO4	Project presentation, software pitch, role play, portfolio preparation, peer review	Apply (C3), Evaluate (C5), Create (C6)	Capstone presentation, professional portfolio, viva voce, peer assessment, faculty rubric

Unit and Co Mapping

Unit No.	Unit Title	CO1	CO2	CO3	CO4
Unit I	Professional Communication for Computing	3			
Unit II	Teamwork and Workplace Effectiveness		3	2	
Unit III	Professionalism, Ethics and Career Readiness	1	1	3	3
Unit IV	Professional Skills	2	2	2	3

	Workshop				
--	----------	--	--	--	--

Recommended Textbooks

1. Meenakshi Raman & Sangeeta Sharma, *Technical Communication: Principles and Practice*, Oxford University Press.
2. Barun K. Mitra, *Personality Development and Soft Skills*, Oxford University Press.
3. Dale Carnegie, *How to Win Friends and Influence People*.
4. Stephen R. Covey, *The 7 Habits of Highly Effective People*.

Reference Books

1. Travis Bradberry & Jean Greaves, *Emotional Intelligence 2.0*.
2. Carnegie Mellon Software Engineering Institute resources on professional software engineering.
3. ACM Code of Ethics and Professional Conduct.
4. IEEE Code of Ethics.

Programme: MCA CBCS– Revised Syllabus w.e.f.-Year 2026 –2027

Semester	Course Code	Course Title	
I	108	Organizational Behavior	
	Prepared by	Dr. Amarja Nargunde	
Type of Course	Credits	Evaluation	Marks
CC	2	IE(50)	50

Course Objectives:

- Understand individual, group, and organizational behavior in technology-driven workplaces.
- Develop skills for effective teamwork, communication, and leadership in IT organizations.
- Analyze motivation, organizational culture, and change management in software and digital enterprises.
- Apply organizational behavior concepts to improve productivity, innovation, and employee well-being.

After completing the course the students shall be able to-

CO1 – Explain key concepts of organizational behavior and their relevance to IT organizations.

CO2 – Analyze individual and team behavior in software development and project environments.

CO3 – Apply leadership, motivation, and communication techniques to improve organizational effectiveness.

CO4 – Evaluate organizational culture, change management, and emerging workplace trends in the IT industry.

Unit	Content	Sessions	COs Number	Teaching Methodology	Cognition Level	Evaluation Tools
1	Introduction to Organizational Behavior; OB models; Individual behavior; Personality, values, attitudes; Emotional intelligence; Diversity and inclusion in IT workplaces.	12	CO1	Lectures, Case Studies	Understand	Quiz, Assignment
2	Perception and learning; Motivation theories (Maslow, Herzberg, Vroom); Employee engagement; Managing knowledge workers and software professionals.	12	CO1 CO2	Discussion, Industry Cases	Apply	Test, Presentation
3	Group dynamics; Team building; Agile and Scrum teams; Communication in distributed teams; Conflict and negotiation in software projects.	12	CO2	Workshops, Simulations	Analyze	Case Analysis, Group Activity
4	Leadership theories; Leadership in technology organizations; Power and politics; Decision-making; Managing virtual and hybrid teams.	12	CO3	Seminars, Role Play	Evaluate	Presentation, Mid-term
5	Organizational culture; Change management; Digital transformation; Innovation and creativity; Future of work, AI impact, ethics and employee well-being.	12	CO4	Case Studies, Project	Create/Evaluate	Project, Viva

Reference Books:

Sr.No.	Name of the Author	Title of the Book	Year Edition	Publisher Company
1	Stephen P. Robbins & Timothy A. Judge	Organizational Behavior	2022, 18th	Pearson
2	Fred Luthans	Organizational Behavior	2023, 14th	McGraw-Hill
3	Parikh & Gupta	Organizational Behaviour	2021, Latest	McGraw-Hill
4	Rao V.S.P.	Organizational Behaviour	2022, Latest	Taxmann
5	Jennifer George & Gareth Jones	Understanding and Managing Organizational Behavior	2021, Latest	Pearson

Online Resources:

Online Resources No.	Website address
1	https://www.shrm.org
2	https://www.gallup.com/workplace
3	https://hbr.org/topic/organizational-behavior

MOOCs:

Resources No.	Website address
1	https://www.coursera.org
2	https://www.edx.org
3	https://onlinecourses.nptel.ac.in