



Anekant Education Society's

**Tuljaram Chaturchand College of Arts, Science & Commerce,
Baramati**

(Empowered Autonomous)

Three/Four Year Honours/Honours with Research BCA Degree

Program in BCA

(Faculty of Commerce and Management)

CBCS Syllabus

TYBCA, Sem.-V

For Department of Computer Application (BCA)

NEP-2.0

Choice Based Credit System Syllabus

(2024 Pattern)

(As Per NEP-2020)

To be Implemented from Academic Year 2026-2027

Title of the Programme: TYBCA**Preamble**

AES's Tuljaram Chaturchand College has decided to change the syllabus of various faculties from June, 2023 by taking into consideration the guidelines and provisions given in the National Education Policy (NEP), 2020. The NEP envisions making education more holistic and effective and to lay emphasis on the integration of general (academic) education, vocational education and experiential learning. The NEP introduces holistic and multidisciplinary education that would help to develop intellectual, scientific, social, physical, emotional, ethical and moral capacities of the students. The NEP 2020 envisages flexible curricular structures and learning based outcomes for the development of the students. The credit structure and the courses framework provided in the NEP are nationally accepted and internationally comparable.

In response to the rapid advancements in science and technology and the evolving approaches in various domains of Computer Science and related subjects, Board of Studies in BCA of Tuljaram Chaturchand College, Baramati - Pune has prepared the syllabus of SYBCA Semester - III under the Choice Based Credit System (CBCS) by following the guidelines of NEP 2020, NCrF, NHEQF, Prof. R.D. Kulkarni's Report, GR of Gov. of Maharashtra dated 20th April, 16th May 2023 and 13th March, 2024 and Circular of SPPU, Pune dated 31st May 2023 and 2nd May, 2024.

BCA is Undergraduate Degree Program with Computer Applications. This program provides sound knowledge of theory and practical's. The different subject helps the students to design, develop and implement software Applications, to learn emerging computer technologies and produce skilled human resource to face the professional challenges.

Overall, revising the BCA syllabus in accordance with the NEP 2020 ensures that students receive an education that is relevant, comprehensive, and prepares them to navigate the dynamic and interconnected world of today. It equips them with the knowledge, skills, and competencies needed to contribute meaningfully to society and pursue their academic and professional goals in a rapidly changing global landscape.

**Programme Outcome for NEP 2020
(With Effect from June 2026-27)
Commerce and Management (Under Graduate Programme)**

PO1: Broad Knowledge and Integrated Learning

Learners will develop wide-ranging knowledge across multiple disciplines and understand how these areas connect with professional fields such as Banking, Accounting, Management, Logistics, Marketing, Human Resource Management, and Computer Science.

PO2: Procedural Understanding for Skill Building

Learners will gain structured and systematic procedural understanding of each subject, enabling mastery of concepts and continuous improvement of practical and academic skills.

PO3: Logical Thinking and Problem-Solving Capability

Learners will be equipped to identify, analyze, and resolve problems in both predictable and unfamiliar situations by applying learned concepts to real-life scenarios. They will also be able to investigate complex problems through experimentation, data evaluation, and interpretation to derive meaningful conclusions.

PO4: Professional and Technical Communication Skills

Learners will enhance verbal and written communication abilities through the effective use of multiple languages. They will be able to express ideas clearly, present reasoned arguments using discipline-specific terminology, participate in constructive discussions, and demonstrate effective communication skills relevant to business and banking environments.

PO5: Analytical Judgment and Decision-Making Skills

Learners will develop the ability to critically examine situations, determine their accuracy and relevance, and make informed decisions. They will analyze business operations and assess the benefits and limitations of trade-related activities using analytical reasoning.

PO6: Creativity, Employability, and Entrepreneurial Mindset

Learners will be capable of recognizing opportunities and converting them into productive outcomes that contribute to personal success and societal advancement. They will be prepared for professional employment, entrepreneurial initiatives, and ethical leadership roles.

PO7: Interdisciplinary Awareness and Competence

Learners will acquire an understanding of principles and values from diverse disciplines and view knowledge as holistic, interconnected, and environmentally responsible, promoting empathy, collaboration, and interdisciplinary problem-solving.

PO8: Ethical Values through Social Participation

Learners will apply their knowledge and attitudes to uphold constitutional, moral, and humanistic values. They will actively engage in community-oriented activities aimed at social development and public well-being.

PO9: Integration of Indigenous Knowledge with Contemporary Practice

Learners will understand traditional knowledge systems and effectively adapt and apply them within modern professional and technological contexts.

PO10: System Planning and Solution Development

Learners will be able to conceptualize, design, and implement solutions for complex real-world computing problems. They will develop systems and processes that address specific needs while considering safety, cultural, social, and environmental responsibilities.

PO11: Ethical Awareness and Social Accountability

Learners will demonstrate an understanding of ethical principles and apply them responsibly in professional and societal contexts. They will exhibit integrity, accountability, respect for societal norms, and adherence to professional ethical standards.

PO12: Research Competence and Methodological Skills

Learners will acquire knowledge of fundamental research methods, processes, and ethical practices necessary for conducting research across various disciplines, including academic, personal, and social contexts.

PO13: Collaborative and Team-Based Skills

Learners will be able to function efficiently and respectfully within team environments, contributing constructively toward shared objectives.

PO14: Specialized Knowledge Application

Learners will demonstrate the ability to apply subject-specific concepts, theories, and models in areas such as Accounting, Taxation, Marketing, Finance, Human Resource Management, and Computer Applications through in-depth subject understanding.

PO15: Environmental Responsibility and Risk Management

Learners will be able to identify environmental risks associated with organizational activities and implement appropriate measures to monitor, control, and minimize impacts on air, water, and soil quality.

Programme Specific Outcomes (PSOs)

- PSO1. Knowledge:** To understand and apply the fundamental principles, concepts, and methods in diverse areas of computer science, computer applications, mathematics, statistics, etc.
- PSO2. Problem Analysis:** Identify, analyze and formulate complex real-life computing problems. Attain substantiated conclusions to solve the problems using fundamental principles of computer science and application domains by using various tools and emerging technologies.
- PSO3. Design and Development:** Design and develop efficient solutions for complex real-world computing problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety and the cultural, societal, and environmental considerations.
- PSO4. Conduct investigations of complex problems:** Ability to research, analyze and Investigate complex computing problems through the design of experiments, analysis, and interpretation of data, and synthesis of the information to arrive at valid conclusions.
- PSO5. Modern Tool Usage:** Create, identify and apply appropriate techniques, skills, and modern computing tools to computing activities.
- PSO6. Ethics and Social Responsibility:** Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.
- PSO7. Individual and Team Work:** Ability to work effectively as an individual, and as a member or leader as per need in, multidisciplinary teams.

PSO8. Life-Long Learning: Recognize the need and have the ability to engage in Independent continuous reflective learning in the context of technological advancement.

PSO9. Project Management: Understand and apply computing, management principles to manage projects.

PSO10. Communication: Able to use interpersonal skills and communicate effectively with the professionals and with society to convey technical information effectively and accurately and able to comprehend and write effective reports, design documentation, and make effective presentations.

PSO11. Innovation, employability, and Entrepreneurial skills: Identify opportunities, and pursue those opportunities to create value and wealth for the betterment of the individual and society at large.

Anekant Education Society's
Tuljaram Chaturchand College, Baramati
(Empowered Autonomous)

Board of Studies in B.C.A.

(Academic Year 2025-26 to 2027-28)

Sr. No.	Name of Member	Designation
1.	Mr. Shah Vishal V.	Chairperson
2.	Mr. Shah Rahul A.	Member
3.	Ms. Kulkarni Prajakta P.	Member
4.	Mrs. Bhagat Asmita A.	Member
5.	Ms. Londhe Kalyani W.	Member
6.	Mrs. Swami Poornima C.	Member
7.	Mr. Chemte Swapnil P.	Member
8.	Ms. Attar Naziya S.	Member
9.	Dr. Ponde Poonam	Vice-Chancellor Nominee Subject Expert from SPPU, Pune
10.	Dr. Dongardive Jyotshna	Subject Expert from Outside the Parent University
11.	Dr. Magre Ritesh	Subject Expert from Outside the Parent University
12.	Mr. Jain Abhishek	Representative from industry/corporate sector/allied areas
13.	Ms. Giramkar Tanuja	Invited –Member
14.	Ms. Kumbhar Trupti	Invited –Member
15.	-	Member of the College Alumni
16.	Mr. Mane Sandip M.	UG Student
17.	-	PG Student

Credit distribution Structure of B.C.A. (2024 Pattern)

Level	Sem.	Core Courses				Minor	GE/OE	AEC	IKS Gen.	VEC	SEC	CC	Total
4.5/100	I	6(T)+6(P)				-	2 (T)	2(T)	2(T)	2(T)	2(P)	-	22
	II	6(T)+6(P)				-	2 (P)	2(T)	-	2(T)	2(P)	2	22
Exit Option : Award of UG Certificate in Major With Total Credits 44 Continue option: Student will select Computer as major and one as Minor.													
Level	Sem.	Credit Related to Major				Minor	GE/OE	AEC	IKS Gen.	VEC	SEC	CC	Total
		Major Core	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(T)	22
	IV	4(T) + 2(P)	-	2 (T/P)	2(CEP)	2(T)+2(P)	2(P)	2(T)	-	-	2(T/P)	2(T)	22
Exit Option: Award of UG Diploma in Major and Minor With Total Credits 88 OR Continue with Major and Minor.													
5.5/300	V	8(T) + 4(P)	2(T) + 2(P)	-	4(OJT)	2(T)	-	-	-	-	-	-	22
	VI	8(T) + 4(P)	2(T) + 2(P)	4 (T/P)	2 (FP)	-	-	-	-	-	-	-	22
Total 3 Years		86				10	08	08	04	04	06	06	132
Exit Option: Award of UG Degree in Major and Minor With Total Credits 132 OR Continue with Major and Minor.													
6.0/400	VII	6(T) + 4(P)	2(T) +2(T/P)	-	4(RP)	4(RM)(T)	-	-	-	-	-	-	22
	VIII	6(T) + 4(P)	2(T) +2(T/P)	-	8(RP)	-	-	-	-	-	-	-	22
Total 4 Years		126				14	08	08	04	04	06	06	176
Four Year UG Honours with Research Degree in Major and Minor with Total credits 176													
6.0/400	VII	10(T) + 4(P)	2(T) +2(T/P)	-	-	4(RM)(T)	-	-	-	-	-	-	22
	VIII	10(T) + 4(P)	2(T) +2(T/P)	-	4(OJT)	-	-	-	-	-	-	-	22
Total 4 Years		126				14	08	08	04	04	06	06	176
Four Year UG Honours Degree in Major and Minor with Total credits 176													

T = Theory, **P** = Practical, **DSC** = Discipline Specific Course, **OE** = Open Elective, **SEC** = Skill Enhancement Course,

IKS = Indian Knowledge System, **AEC** = Ability Enhancement Course, **VEC** = Value Education

Course Structure of F.Y. BCA (2024 Pattern)
(As Per NEP-2020)

FY BCA – Semester –I							
Course Type	Course Code	Paper Title	Hours/ Week	Credits	Internal	External	Total
Major Mandatory	BCA-101 GEN	Problem Solving Techniques and Basic C Programming	02 (T)	02	20	30	50
	BCA-102 GEN	Lab Course on BCA-101 GEN	04(P)	02	25	25	50
	BCA-103 GEN	Computer Architecture	02(T)	02	20	30	50
	BCA-104 GEN	Lab Course on BCA-103 GEN	04(P)	02	25	25	50
	BCA-105 GEN	Foundation of Mathematics for Computer Science	02(T)	02	20	30	50
	BCA-106 GEN	Lab Course on BCA-105 GEN	04(P)	02	25	25	50
Open Elective (OE)	BBA-104 OE	Introduction to Data Science	02(T)	02	20	30	50
Skill Enhancement Course (SEC)	BCA-101 SEC	HTML & Web Page Designing	04(P)	02	25	25	50
IKS Generic	GEN-106-IKS	Indian Knowledge System	02(T)	02	20	30	50
Ability Enhancement Course (AEC)	ENG-101 AEC	General English – I	02(T)	02	20	30	50
Value Education Course (VEC)	ENV-105 VEC	Environmental Awareness	02(T)	02	20	30	50
Total				22	240	310	550

FY BCA – Semester -II							
Course Type	Course Code	Paper Title	Hours/Week	Credits	Internal	External	Total
Major Mandatory	BCA-151 GEN	Advanced C Programming	02 (T)	02	20	30	50
	BCA-152 GEN	Lab Course on BCA-151 GEN	04(P)	02	25	25	50
	BCA-153 GEN	Introduction to Microcontroller	02(T)	02	20	30	50
	BCA-154 GEN	Lab Course on BCA-153 GEN	04(P)	02	25	25	50
	BCA-155 GEN	Linear Algebra	02(T)	02	20	30	50
	BCA-156 GEN	Lab Course on BCA-155 GEN	04(P)	02	25	25	50
Open Elective (OE)	BBA-154 OE	Data Science Using Spread Sheet	04(P)	02	25	25	50
Skill Enhancement Course (SEC)	BCA-151 SEC	Software Tools for Business Communication	04(P)	02	25	25	50
Ability Enhancement Course (AEC)	ENG-151 AEC	General English – II	02(T)	02	20	30	50
Value Education Course (VEC)	COS-155 VEC	Digital Technological Solution	02(T)	02	20	30	50
Co-curricular Course (CC)	PES-156 CC	Physical Education	02	02	20	30	50
Total				22	245	305	550

Course Structure of S.Y. BCA (2024 Pattern)
(As Per NEP-2020)

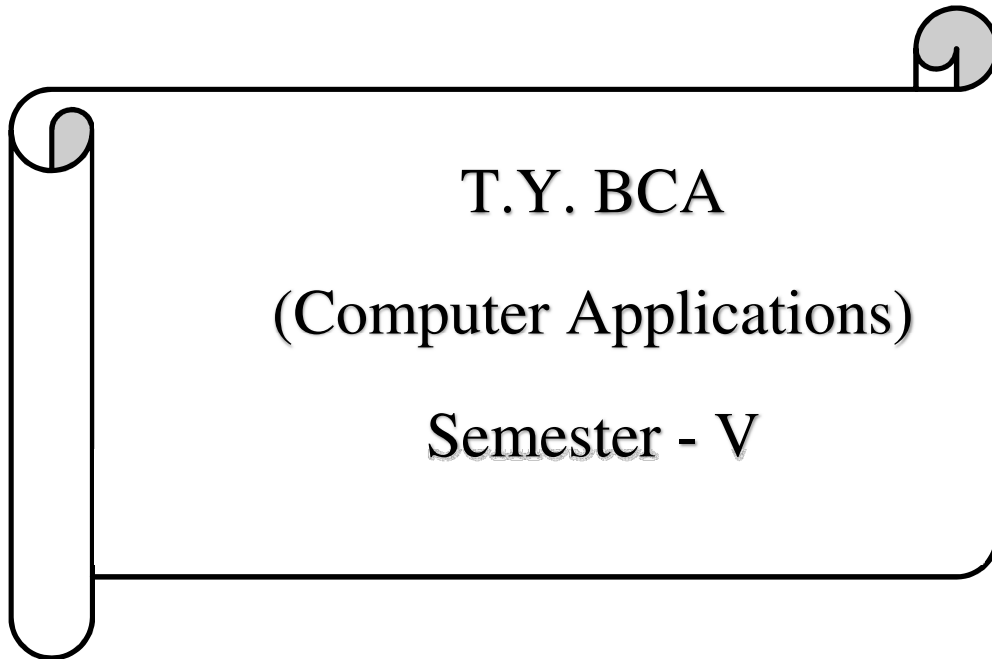
SY BCA – Semester -III				
Course Type	Course Code	Course Title	Theory/ Practical	Credits
Major Mandatory	BCA-201-MRM	Algorithms & Data Structures	Theory	02
	BCA-202-MRM	Database Management Systems	Theory	02
	BCA-203-MRM	Practical Lab on Algorithms & Data Structures	Practical	02
Vocational Skill Course (VSC)	BCA-204-VSC	Practical Lab on Database Management Systems	Practical	02
Field Project(FP)	BCA-205-FP	Field Project	Practical	02
Minor	BCA-206-MN	Fundamentals of ICT	Theory	02
	BCA-207-MN	Practical Lab on Advanced Excel	Practical	02
Open Elective (OE)	BCA-208-OE	Informatics and Cyber laws	Theory	02
Subject Specific Indian Knowledge System (IKS)	BCA-209-IKS	Ancient Indian Foundations of Computing	Theory	02
Ability Enhancement Course (AEC)	MAR-210-AEC/ HIN-210-AEC/SAN-210-AEC	Selected from AEC Basket	Theory (Any One)	02
Co-curricular Course (CC)	YOG/PES/CUL/ N SS/NCC-211-CC	To be continued from the Semester – II		02
Total Credits of Semester - III				22

SY BCA – Semester –IV				
Course Type	Course Code	Course Title	Theory/ Practical	Credits
Major Mandatory	BCA-251-MRM	Advanced Data Structures	Theory	02
	BCA-252-MRM	Relational Database Management System	Theory	02
	BCA-253-MRM	Practical Lab on Advanced Data Structures	Practical	02
Vocational Skill Course (VSC)	BCA-254-VSC	Object Oriented Programming using Java	Theory	02
Community Engagement Project (CEP)	BCA-255-CEP	Community Engagement Project	Practical	02
Minor	BCA-256-MN	Digital Marketing	Theory	02
	BCA-257-MN	Practical Lab on Digital Marketing	Practical	02
Open Elective (OE)	BBA-257-OE	Hands on Google Apps	Practical	02
Skill Enhancement Course (SEC)	BCA-259-SEC	Practical Lab on RDBMS	Practical	02
Ability Enhancement Course (AEC)	MAR-260-AEC/ HIN-260-AEC/ SAN-260-AEC	To be continued from the Semester – III	Theory (Any One)	02
Co-curricular Course (CC)	YOG/PES/CU L/ NSS/NCC- 261- CC	To be continued from the Semester – III		02
Total Credits of Semester - IV				22
Total Credits Semester – III + IV				44

Course Structure of T.Y. BCA (2024 Pattern)
(As Per NEP-2020)

TY BCA – Semester –V				
Course Type	Course Code	Course Title	Theory/ Practical	Credits
Major Mandatory	BCA-301-MRM	Advanced Java	Theory	02
	BCA-302-MRM	Basic Python Programming	Theory	02
	BCA-303-MRM	Computer Network	Theory	02
	BCA-304-MRM	Software Engineering	Theory	02
Major Mandatory	BCA-305-MRM	Lab Course On BCA-301	Practical	02
	BCA-306-MRM	Lab Course On BCA-302	Practical	02
Major Elective (MJE)	BCA-307-MJE(A)	Operating System	Theory (Any One)	02
	BCA-308-MJE(A)	Block Chain Technology		
	BCA-307-MJE(B)	Lab Course On BCA-307- MJE(A)	Practical (Any One)	02
	BCA-308-MJE(B)	Lab Course On BCA-308- MJE(B)		
On Job Training (OJT)	BCA-309-OJT	On Job Training	Practical	04
Minor	BCA-310-MN	E-Commerce	Theory	02
Total Credits of Semester - V				22

TY BCA – Semester –VI				
Course Type	Course Code	Course Title	Theory/ Practical	Credits
Major Mandatory	BCA-351-MRM	Advanced Python Programming	Theory	02
	BCA-352-MRM	PHP Programming Essentials	Theory	02
	BCA-353-MRM	Cryptography	Theory	02
	BCA-354-MRM	Introduction to Project Management	Theory	02
Major Mandatory	BCA-355-MRM	Lab Course On BCA-351	Practical	02
	BCA-356-MRM	Lab Course On BCA-352	Practical	02
Major Elective (MJE)	BCA-357-MJE(A)	Software Testing and Quality Assurance	Theory (Any One)	02
	BCA-358-MJE(A)	Cyber Security and Ethical Hacking		
	BCA-357-MJE(B)	Lab Course On BCA-357-MJE(A)	Practical (Any One)	02
	BCA-358-MJE(B)	Lab Course On BCA-358-MJE(B)		
Vocational Skill Course (VSC)	BCA-359-VSC	Emerging Technologies in Business	Theory	02
	BCA-360-VSC	Lab course on Data Analytics Using Power BI/ Tableau	Practical	02
Field Project(FP)	BCA-361-FP	Field Project	Practical	02
Total Credits of Semester - VI				22
Total Credits of Semester – V+VI				44



CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science) (2024 Pattern)

Name of the Programme	: B.C.A. (Science)
Program Code	: BCA
Class	: T.Y.B.C.A
Semester	: V
Course Type	: Major Mandatory [Theory]
Course Name	: Advanced Java
Course Code	: BCA-301-MRM
No. of Credits	: 02
No. of Teaching Hours	: 30

Prerequisites:

Understanding of Problem solving techniques using a programming language and basic data structures.

Course Objectives:

1. To understand and apply core concepts of the Java Collections Framework and object-oriented programming.
2. To develop GUI-based and multithreaded Java applications using AWT and Swing.
3. To implement exception handling, file handling, and database connectivity using JDBC.
4. To design and develop dynamic web applications using Servlets and JSP.
5. To integrate Java technologies to build secure, efficient, and real-world software solutions with professional and ethical responsibility.

Course Outcomes:

At the end of the course, students will be able to:

- CO1:** Apply concepts of the Java Collections Framework to manage and manipulate data efficiently.
- CO2:** Design and develop GUI-based applications using AWT and Swing components.
- CO3:** Implement exception handling, multithreading, and file handling to build robust Java applications.
- CO4:** Develop database-driven applications using JDBC for data storage and retrieval.
- CO5:** Build dynamic web applications using Servlets and JSP.
- CO6:** Apply networking concepts to develop distributed Java applications.
- CO7:** Integrate multithreading, GUI, event handling, and database connectivity to design real-world Java systems with ethical and professional responsibility.

Topics and Learning Points

Unit	Title and Contents	No. of Lectures
Unit 1	Collections Framework <ul style="list-style-type: none"> • Wrapper Classes • Introduction to the Collection framework • List – Array List, Linked List and Vector, Stack, Queue • Set – Hash Set, Tree Set, and Linked Hash Set • Map – Hash Map, Linked Hash Map, Hash table and Tree • Map Interfaces such as Comparator, Iterator, List Iterator, Enumeration 	06
Unit 2	Multithreading <ul style="list-style-type: none"> • What are threads? • Life cycle of Thread • Running and starting thread using class and interface • Thread priorities • Running multiple threads • synchronization and inter thread communication 	10
Unit 3	User Interface Components with Swing <ul style="list-style-type: none"> • What is Swing? • Garbage Collection • The MVC Architecture and Swing • Layout Manager and Layouts, The JComponent class • Different components from Swing • JLabel. JButton. JText .JTextArea. JCheckBox. JRadioButton. • JList. JComboBox.JMenu And JPopupMenu Class, JMenuItem • Dialogs (Message, confirmation, input), JFileChooser, JColorChooser • Event Handling: Event sources, Listeners • Mouse and Keyboard Event Handling • Adapters – MouseAdapter ,KeyAdapter • Lambda Function 	08
Unit 4	Database Programming <ul style="list-style-type: none"> • The Role of JDBC, JDBC configuration • Types of Drivers • Connectivity with Database • Executing SQL statements, query execution 	06

	<ul style="list-style-type: none"> • Scrollable and updatable result sets • TYPE_FORWARD_ONLY , TYPE_SCROLL_INSENSITIVE ,TYPE_SCROLL_SENSITIVE • Metadata – DatabaseMetadata, ResultSetMetadata • Transactions – commit(), rollback(), SavePoint 	
--	--	--

Reference /Text Books:

Text Books:

1. Herbert Schildt, *Java: The Complete Reference*, McGraw-Hill Education.
2. Cay S. Horstmann, *Core Java Volume II – Advanced Features*, Pearson Education.
3. Ivor Horton, *Beginning Java 2 SDK 1.4 Edition*, Wiley India.
4. E. Balagurusamy, *Programming with Java*, McGraw-Hill Education.

Reference Books:

1. Effective Java – Joshua Bloch – Best practices and advanced concepts for writing robust and maintainable Java applications.
2. Head First Java – Kathy Sierra & Bert Bates – Beginner-friendly approach with visually rich explanations.
3. Java: A Beginner’s Guide – Herbert Schildt – Clear explanations and practical examples for learners.
4. Java Concurrency in Practice – Brian Goetz – Comprehensive guide to multithreading and concurrency in Java.
5. Clean Code: A Handbook of Agile Software Craftsmanship – Robert C. Martin – Principles for writing clean, maintainable, and professional code.

Web Resources:

1. W3Schools – Java Tutorial
<https://www.w3schools.com/java/>
2. GeeksforGeeks – Java Programming and JDBC Tutorials
<https://www.geeksforgeeks.org/java/>
3. Tutorialspoint – Advanced Java Tutorials
<https://www.tutorialspoint.com/java/>
4. JavaTpoint – Servlets, JSP, and JDBC Tutorials
<https://www.javatpoint.com/>

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PO13	PO14	PO15
CO1	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
CO2	2	3	2	1	2	2	1	1	1	2	1	1	1	1	1
CO3	2	3	3	2	2	3	2	1	1	3	1	2	2	2	2
CO4	1	3	2	3	2	2	2	1	1	2	1	1	2	1	3
CO5	2	2	3	2	3	2	2	2	1	3	3	2	1	2	2
CO6	2	3	3	2	3	3	2	2	2	3	2	2	3	1	2
CO7	3	2	3	2	3	3	3	2	2	2	2	3	2	2	1

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

PO1: Broad Knowledge and Integrated Learning

CO1: Java Collections Framework strengthens foundational and integrated programming knowledge.

CO2: GUI development using AWT and Swing enhances interdisciplinary technical understanding.

CO3: Exception handling, multithreading, and file handling integrate core system concepts.

CO4: JDBC connects programming with database management systems.

CO5: Servlets and JSP integrate web technologies with backend systems.

CO6: Networking concepts broaden knowledge of distributed systems.

CO7: Integration of GUI, threading, and database connectivity provides comprehensive system-level understanding.

Mapping of PO2 with All CO'S

PO2: Procedural Understanding for Skill Building

CO1: Implementing List, Set, and Map builds structured programming skills.

CO2: Designing GUI applications enhances systematic development practices.

CO3: Multithreading and exception handling strengthen procedural coding techniques.

CO4: JDBC implementation develops structured database interaction skills.

CO5: Web application development improves enterprise-level procedural practices.

CO6: Networking implementation strengthens protocol-based programming skills.

CO7: Integrating multiple technologies enhances advanced implementation capability.

Mapping of PO3 with All CO'S

PO3: Logical Thinking and Problem-Solving Capability

- CO1:** Choosing appropriate collection types enhances logical decision-making.
- CO2:** GUI event handling improves analytical thinking.
- CO3:** Thread synchronization and exception handling develop problem-solving skills.
- CO4:** SQL query execution enhances logical data manipulation ability.
- CO5:** Web application logic improves real-world problem-solving capability.
- CO6:** Networking troubleshooting enhances critical thinking skills.
- CO7:** Designing integrated systems strengthens comprehensive solution-building ability.
-

Mapping of PO4 with All CO'S

PO4: Professional and Technical Communication Skills

- CO1:** Organized collection usage improves structured coding presentation.
- CO2:** GUI interfaces improve user interaction and technical communication.
- CO3:** Proper error messages enhance system-user communication.
- CO4:** Database reporting improves structured information representation.
- CO5:** Web applications enhance interactive communication platforms.
- CO6:** Networking supports communication between systems.
- CO7:** Integrated applications improve overall system documentation and clarity.
-

Mapping of PO5 with All CO'S

PO5: Analytical Judgment and Decision-Making Skills

- CO1:** Selecting suitable data structures improves analytical judgment.
- CO2:** GUI design decisions enhance evaluation skills.
- CO3:** Managing threads and exceptions strengthens analytical debugging.
- CO4:** Choosing proper database operations improves decision-making.
- CO5:** Web architecture selection enhances system optimization skills.
- CO6:** Networking configuration builds analytical assessment ability.
- CO7:** Integrating technologies requires high-level evaluation and decision-making.
-

Mapping of PO6 with All CO'S

PO6: Creativity, Employability, and Entrepreneurial Mindset

- CO1:** Collections are widely used in enterprise applications.
- CO2:** GUI development enhances employability in desktop application design.
- CO3:** Multithreading improves performance-oriented professional development.
- CO4:** JDBC knowledge supports backend development careers.
- CO5:** Servlets and JSP enable full-stack web development opportunities.
- CO6:** Networking skills enhance career prospects in distributed systems.
- CO7:** Integrated real-world systems foster innovative and entrepreneurial thinking.
-

Mapping of PO7 with All CO'S

PO7: Interdisciplinary Awareness and Competence

- CO1:** Collections manage data across business and scientific domains.
- CO2:** GUI applications apply to banking, management, and logistics systems.
- CO3:** File handling supports cross-domain data management.
- CO4:** Database programming integrates business information systems.
- CO5:** Web applications connect multiple organizational domains.

CO6: Networking supports multidisciplinary communication systems.

CO7: Integrated Java systems support enterprise-level interdisciplinary solutions.

Mapping of PO8 with All CO'S

PO8: Ethical Values through Social Participation

CO1: Proper data structure usage promotes responsible software design.

CO2: User-friendly GUI ensures ethical usability.

CO3: Exception handling prevents misuse and system failure.

CO4: Secure database connectivity ensures data privacy.

CO5: Web security promotes responsible digital participation.

CO6: Networking security ensures ethical data transmission.

CO7: Professional system integration encourages responsible computing.

Mapping of PO9 with All CO'S

PO9: Integration of Indigenous Knowledge with Contemporary Practice

CO1: Traditional data organization methods are implemented digitally using collections.

CO2: GUI systems digitize manual processes.

CO3: Threading enhances traditional processing efficiency in digital form.

CO4: Databases digitize conventional record-keeping systems.

CO5: Web applications modernize traditional service delivery.

CO6: Networking connects traditional systems with modern infrastructure.

CO7: Integrated Java systems transform conventional workflows into digital platforms.

Mapping of PO10 with All CO'S

PO10: System Planning and Solution Development

CO1: Collections support backend system architecture.

CO2: GUI design forms system interface planning.

CO3: Multithreading enhances scalable system design.

CO4: JDBC manages system-level data storage.

CO5: Servlets and JSP form web-based system solutions.

CO6: Networking ensures distributed system deployment.

CO7: Integration of all components supports complete system development.

Mapping of PO11 with All CO'S

PO11: Ethical Awareness and Social Accountability

CO1: Clean coding practices promote professional responsibility.

CO2: GUI usability ensures inclusive system design.

CO3: Exception handling prevents data corruption.

CO4: Secure database transactions ensure accountability.

CO5: Responsible web design ensures ethical data handling.

CO6: Networking security supports accountable communication.

CO7: Integrated systems emphasize ethical and professional responsibility.

Mapping of PO12 with All CO'S

PO12: Research Competence and Methodological Skills

- CO1: Collections support algorithm experimentation.
 - CO2: GUI prototypes support research-based applications.
 - CO3: Threading improves performance experimentation.
 - CO4: JDBC supports data-driven research systems.
 - CO5: Web applications assist research dissemination.
 - CO6: Networking supports distributed research environments.
 - CO7: Integrated Java systems enhance applied research capability.
-

Mapping of PO13 with All CO'S

PO13: Collaborative and Team-Based Skills

- CO1: Collection-based projects support team coding tasks.
 - CO2: GUI development can be divided among team members.
 - CO3: Multithreading projects require collaborative debugging.
 - CO4: Database projects involve backend team coordination.
 - CO5: Web applications promote collaborative development.
 - CO6: Networking configuration requires teamwork.
 - CO7: Integrated systems development enhances team coordination skills.
-

Mapping of PO14 with All CO'S

PO14: Specialized Knowledge Application

- CO1: Advanced collection usage strengthens Java specialization.
 - CO2: GUI development builds subject-specific expertise.
 - CO3: Multithreading enhances advanced Java knowledge.
 - CO4: JDBC develops backend specialization.
 - CO5: Servlets and JSP enhance enterprise-level expertise.
 - CO6: Networking strengthens distributed system specialization.
 - CO7: Integration of technologies builds comprehensive Java mastery.
-

Mapping of PO15 with All CO'S

PO15: Environmental Responsibility and Risk Management

- CO1: Efficient data structures reduce memory wastage.
 - CO2: Optimized GUI design minimizes resource consumption.
 - CO3: Controlled multithreading reduces system overload risks.
 - CO4: Transaction management prevents data loss risks.
 - CO5: Digital web applications reduce paper-based processes.
 - CO6: Secure networking reduces cyber-security risks.
 - CO7: Integrated system design ensures sustainable and risk-aware software solutions.
-

CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science) (2024 Pattern)

Name of the Programme	: B.C.A. (Science)
Program Code	: BCA
Class	: T.Y.B.C.A
Semester	: V
Course Type	: Major Mandatory [Theory]
Course Name	: Basic Python Programming
Course Code	: BCA-302-MRM
No. of Credits	: 02
No. of Teaching Hours	: 30

Prerequisites:

Understanding of Problem solving techniques using a programming language and basic data structures.

Course Objectives:

1. Develop modular Python programs.
2. Apply suitable Python programming constructs, built-in data structures using Python libraries to solve a problem.
3. Implement functions and modules to enhance code reusability and maintainability
4. Demonstrate proficiency in handling different data structures such as lists, tuples, and dictionaries in Python.
5. Understand basic File handling in Python.
6. Apply exception handling mechanisms to build robust and error-free Python programs.
7. Utilize Python libraries for data visualization and mathematical computations.

Course Outcomes:

At the end of the course, students will be able to:

- CO1:** Develop modular Python programs.
- CO2:** Apply Python constructs and built-in data structures.
- CO3:** Implement functions and modules for reusability.
- CO4:** Demonstrate proficiency in lists, tuples, sets, and dictionaries.
- CO5:** Understand basic file handling in Python.
- CO6:** Apply exception handling mechanisms.
- CO7:** Utilize Python libraries for visualization and computations.

Topics and Learning Points

Unit	Title and Contents	No. of Lectures
Unit 1	<p>Introduction to Python Programming</p> <ul style="list-style-type: none"> • What can Python do? • Why Use Python? • History, features and Application areas of Python • Python installation and IDEs • Structure of Python Program • Variables, constants, identifiers and Keywords • Basic Data Types and type conversion • Statements and expressions • Input/output statements 	06
Unit 2	<p>Control Structures and Functions</p> <ul style="list-style-type: none"> • Operators <ul style="list-style-type: none"> ○ Arithmetic ○ Relational ○ Logical ○ Bitwise • Conditional Flow statements <ul style="list-style-type: none"> ○ If ○ if-else ○ nested if • Loops <ul style="list-style-type: none"> ○ For ○ While ○ Do-while ○ nested loops • Continue, break, pass and exit • Function and Its Types <ul style="list-style-type: none"> ○ Built-In Functions ○ User-defined Functions ○ Function Definition and call ○ Scope and Lifetime of Variables ○ Command Line Arguments ○ Recursive Functions ○ Lambda (Anonymous) Functions • Assert statement • Importing User defined module and Packages 	10

Unit 3	Mutable, Immutable objects <ul style="list-style-type: none"> • Lists • Tuples • Set • Dictionaries • Math and Numpy module 	08
Unit 4	Files and Exception Handling <ul style="list-style-type: none"> • Types of Files • Creating, Reading and Writing on Text and Binary files • Reading and Writing CSV Files and JSON files • The PickleModule • Try-except-else-finally block • Raise statement • Hierarchy of exceptions • Adding exceptions 	06

Reference /Text Books:

Text Books :

1. Venkatesh, Nagaraju Y, Introduction to Python Programming, Khanna Publishing House, 2021.
2. Jeeva Jose, Introduction to Computing & Problem Solving With PYTHON, Khanna Publishing House, 2023.
3. Sheetal Taneja & Naveen kumar: Python Programming a Modular approach – A Modular approach with Graphics, Database, Mobile and Web applications, Pearson, 2017.

Reference Books:

1. An introduction to Python for absolute beginners, by Bob Dowling, Cambridge Univ.
2. Introduction to Computation and Programming using Python, by John Guttag, 2nd edition, 2016, PHI India.

Web Resources:

1. <https://www.w3schools.com/python/default.asp>
2. <https://www.learnpython.org>

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO1 12	PO 13	PO 14	PO 15
CO1	3	3	3	3	3	2	2	2	2	3	2	2	2	3	2
CO2	3	3	3	3	3	2	2	2	2	3	2	2	2	3	2
CO3	3	3	3	3	3	3	2	2	2	3	2	2	3	3	2
CO4	3	3	2	2	3	2	2	2	2	2	1	2	2	3	1
CO5	2	2	2	2	2	2	3	2	2	2	2	2	2	3	2
CO6	3	3	3	3	3	2	2	2	2	3	3	3	3	3	2
CO7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

PO1: Broad Knowledge and Integrated Learning

CO1:PO1: Modular programming strengthens fundamental and integrated programming knowledge.

CO2:PO1: Applying constructs and data structures builds strong conceptual understanding.

CO3:PO1: Functions and modules enhance knowledge of structured software development.

CO4:PO1: Data structures improve understanding of data organization principles.

CO5:PO1: File handling develops knowledge of persistent data management systems.

CO6:PO1: Exception handling builds understanding of reliable software mechanisms.

CO7:PO1: Python libraries expand interdisciplinary knowledge in analytics and computation.

Mapping of PO2 with All CO'S

PO2: Procedural Understanding for Skill Building

CO1:PO2: Modular programming promotes systematic coding practices.

CO2:PO2: Structured constructs improve procedural programming skills.

CO3:PO2: Functions enhance organized and reusable code development.

CO4:PO2: Data structures strengthen procedural data handling techniques.

CO5:PO2: File operations build structured data processing skills.

CO6:PO2: Exception handling improves systematic debugging skills.

CO7:PO2: Using libraries enhances advanced procedural implementation skills.

Mapping of PO3 with All CO'S

PO3: Logical Thinking and Problem-Solving Capability

CO1:PO3: Breaking programs into modules enhances logical reasoning.

CO2:PO3: Control structures develop analytical problem-solving ability.

CO3:PO3: Reusable functions promote structured solution design.

CO4:PO3: Data structures improve logical data manipulation.

CO5:PO3: File handling solves real-world storage problems logically.

CO6:PO3: Exception handling improves troubleshooting and critical thinking.

CO7:PO3: Visualization tools support analytical decision-making.

Mapping of PO4 with All CO'S

PO4: Professional and Technical Communication Skills

CO1:PO4: Modular programs improve clarity and documentation.

CO2:PO4: Structured coding enhances technical explanation ability.

CO3:PO4: Functions promote readable and maintainable code.

CO4:PO4: Data structures help present organized solutions.

CO5:PO4: File handling supports structured reporting.

CO6:PO4: Error messages improve communication between user and system.

CO7:PO4: Data visualization improves graphical communication of results.

Mapping of PO5 with All CO'S

PO5: Analytical Judgment and Decision-Making Skills

CO1:PO5: Breaking programs into modules develops analytical judgment.

CO2:PO5: Choosing operators and loops improves decision-making ability.

CO3:PO5: Selecting reusable modules strengthens solution evaluation.

CO4:PO5: Choosing appropriate data structures enhances analytical thinking.

CO5:PO5: File type selection (text, binary, CSV, JSON) builds decision skills.

CO6:PO5: Identifying correct exception types enhances analytical debugging.

CO7:PO5: Graphical data analysis supports informed decision-making.

Mapping of PO6 with All CO'S

PO6: Creativity, Employability, and Entrepreneurial Mindset

CO1:PO6: Modular programs support scalable application development.

CO2:PO6: Control structures enable development of real-world applications.

CO3:PO6: Reusable modules enhance professional coding standards.

CO4:PO6: Data structures support application development in industry.

CO5:PO6: File systems are essential in business applications.

CO6:PO6: Robust error handling improves professional software reliability.

CO7:PO6: NumPy and visualization tools enhance employability in analytics fields.

Mapping of PO7 with All CO'S

PO7: Interdisciplinary Awareness and Competence

CO1:PO7: Python basics apply across domains (banking, management, science).

CO2:PO7: Logical constructs support interdisciplinary software solutions.

CO3:PO7: Modules can be reused in finance, HR, logistics systems.

CO4:PO7: Data structures manage interdisciplinary datasets.

CO5:PO7: CSV and JSON files support cross-domain data exchange.

CO6:PO7: Reliable programs are essential in multidisciplinary systems.

CO7:PO7: NumPy supports mathematical and scientific computations.

Mapping of PO8 with All CO'S**PO8: Ethical Values through Social Participation**

CO1:PO8: Structured coding promotes responsible development.

CO2:PO8: Logical validation reduces system misuse.

CO3:PO8: Reusability reduces resource waste.

CO4:PO8: Responsible data handling ensures ethical data management.

CO5:PO8: Secure file handling ensures privacy protection.

CO6:PO8: Exception handling prevents system failures affecting users.

CO7:PO8: Transparent visualization promotes ethical reporting.

Mapping of PO9 with All CO'S**PO9: Integration of Indigenous Knowledge with Contemporary Practice**

CO1:PO9: Traditional logical methods are implemented through modular programs.

CO2:PO9: Classical algorithms are structured using Python constructs.

CO3:PO9: Functions allow implementation of traditional formulas digitally.

CO4:PO9: Data structures digitize manual record-keeping systems.

CO5:PO9: File handling converts traditional records into digital formats.

CO6:PO9: Risk-handling principles apply to exception management.

CO7:PO9: Statistical methods are implemented using modern libraries.

Mapping of PO10 with All CO'S**PO10: System Planning and Solution Development**

CO1:PO10: Modular programming supports structured system architecture.

CO2:PO10: Control structures form core system logic.

CO3:PO10: Modules enhance scalable system design.

CO4:PO10: Data structures support backend system components.

CO5:PO10: File systems manage system-level data storage.

CO6:PO10: Exception handling ensures stable system performance.

CO7:PO10: Libraries support analytical and computational systems.

Mapping of PO11 with All CO'S**PO11: Ethical Awareness and Social Accountability**

CO1:PO11: Clean modular coding promotes professional ethics.

CO2:PO11: Proper syntax ensures responsible coding practices.

CO3:PO11: Maintainable modules promote sustainable software design.

CO4:PO11: Ethical data handling ensures integrity.

CO5:PO11: Secure file operations protect user data.

CO6:PO11: Exception handling prevents data loss and failures.

CO7:PO11: Ethical data visualization ensures transparency.

Mapping of PO12 with All CO'S**PO12: Research Competence and Methodological Skills**

CO1:PO12: Modular design supports research experiments.

CO2:PO12: Logical constructs implement research algorithms.

CO3:PO12: Functions assist in structured research modeling.

CO4:PO12: Data structures manage research datasets.

CO5:PO12: Files store research data efficiently.

CO6:PO12: Exception handling ensures research result accuracy.

CO7:PO12: NumPy enables statistical and computational research analysis.

Mapping of PO13 with All CO'S

PO13: Collaborative and Team-Based Skills

CO1:PO13: Modular coding enables task distribution.

CO2:PO13: Structured logic improves collaborative development.

CO3:PO13: Reusable modules support team integration.

CO4:PO13: Data management can be team-based.

CO5:PO13: Shared file handling supports teamwork.

CO6:PO13: Debugging encourages team collaboration.

CO7:PO13: Visualization tools support collaborative data analysis.

Mapping of PO14 with All CO'S

PO14: Specialized Knowledge Application

CO1:PO14: Develops expertise in Python programming fundamentals.

CO2:PO14: Strengthens specialization in programming constructs.

CO3:PO14: Enhances advanced software development skills.

CO4:PO14: Builds strong knowledge of data structures.

CO5:PO14: Develops practical computing applications.

CO6:PO14: Improves knowledge of robust system development.

CO7:PO14: Advances specialization in computational and analytical tools.

Mapping of PO15 with All CO'S

PO15: Environmental Responsibility and Risk Management

CO1:PO15: Modular coding reduces redundant processing.

CO2:PO15: Efficient constructs reduce computational waste.

CO3:PO15: Reusable modules minimize development overhead.

CO4:PO15: Optimized data structures reduce memory usage.

CO5:PO15: Digital file systems reduce paper usage.

CO6:PO15: Exception handling prevents system crashes and resource loss.

CO7:PO15: Efficient computational libraries optimize energy usage.

**CBCS Syllabus as per NEP 2020 for T.Y.B.C.A (Science)
(2024 Pattern)**

Name of the Programme	: B.C.A. (Science)
Programme Code	: BCA
Class	: T.Y.B.C.A.
Semester	: V
Course Type	: Major Mandatory
Course Code	: BCA-303-MRM
Course Title	: Computer Network
No. of Credits	: 02
No. of Teaching Hours	: 30

Prerequisites:

Basic knowledge of computer fundamentals, operating systems, and fundamental concepts of data communication.

Course Objectives:

1. To understand the fundamental concepts of computer networks, networking models, and types of networks.
2. To study the OSI and TCP/IP reference models and understands the functions of each layer.
3. To analyze data transmission techniques, transmission media, switching methods, and error detection/correction mechanisms.
4. To understand data link layer protocols, flow control, and medium access control techniques.
5. To examine routing algorithms and congestion control techniques in the network layer.
6. To explore transport layer protocols (TCP/UDP) and their reliability and flow control mechanisms.

Course Outcomes (COs):

At the end of the course, students will be able to:

CO1: Explain basic concepts of data communication, networking criteria, and network categories (LAN, MAN, WAN).

CO2: Identify and compare different network topologies and transmission modes.

- CO3:** Describe digital data transmission methods including parallel and serial transmission, DTE-DCE interface, and modem standards.
- CO4:** Explain the OSI reference model and analyze the functions of each layer.
- CO5:** Compare TCP/IP and OSI models and explain Internet architecture and client-server model.
- CO6:** Demonstrate understanding of IP addressing classes and Internet structure including ARPANET and WWW.
- CO7:** Explain and differentiate core networking protocols such as IP, TCP, HTTP, FTP, and ARP.

Topics and Learning Points

Units	Title & Content	No. of Lectures
I	<p>Introduction to Computer Networking</p> <p>Data Communication, Networks – Distributed Processing, Network Criteria, Applications; Protocols and Standards, Standard Organization, Line Configuration – Point to Point, Multi Point; Topology – Mesh, Star, Tree, Bus, Ring, Hybrid; Transmission mode, Categories of Network – LAN, MAN, WAN, Inter Networks</p>	[06]
II	<p>Transmission of Digital Data</p> <p>Analog and Digital, digital data transmission – parallel transmission, serial transmission, DTE-DCE interface – data terminal equipment, data circuit terminating equipment, standards, modems- Transmission rate, Modem standards.</p>	[06]
III	<p>The OSI Model</p> <p>ISO organization, The model – Layered architecture, functions of the layers -Physical layer, Data Link layer, Network layer, Transport layer, session layer, Presentation layer, Application layer.</p>	[09]
IV	<p>TCP/IP Model & Protocols</p> <p>The TCP/IP reference model, comparison of TCP/IP & OSI, Introduction to Internet – ARPANET, Architecture of Internet, Client server model, www, IP Address Classes, Protocols: IP, HTTP, TCP. FTP, ARP.</p>	[09]

Reference /Text Books:

1. Introduction to Data communication & Networking – Behrouz & Forouzan

2. Computer Networking – Andres & Tanenbaum.
3. Networking All In One Dummies Wiley Publication.[5th Edition]

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PO13	PO14	PO15
CO1	3	1	1	1	1	1	1	1	1	1	1	1	1	2	1
CO2	3	2	2	2	2	2	2	2	1	2	1	2	2	2	1
CO3	3	3	3	2	3	3	3	2	2	3	2	3	3	3	2
CO4	2	3	3	2	3	2	2	2	2	3	2	3	2	3	2
CO5	3	3	3	3	3	3	3	2	2	3	2	3	3	3	2
CO6	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
CO7	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

1. Justification of PO1 to ALL COs:

CO1 - Provides fundamental knowledge of computer network concepts and architecture essential for integrated learning.

CO2 - Develops strong understanding of standard networking models forming the theoretical foundation of networking systems.

CO3 - Enhances analytical knowledge of transmission methods and network structures used in real-world communication.

CO4 - Builds moderate understanding of data link layer operations and error control techniques.

CO5 - Strengthens core knowledge of routing and addressing required for network design and communication.

CO6 - Provides understanding of transport layer services supporting reliable data communication.

CO7 - Introduces integrated knowledge of security and performance management in networks.

2. Justification of PO2 to ALL COs:

CO1 - Provides basic procedural awareness of networking components required for skill development.

CO2 - Helps learners understand structured procedures followed in layered network communication models.

CO3 - Develops strong procedural skills in selecting transmission techniques and network configurations.

CO4 - Enhances practical understanding of protocol operations and error handling mechanisms.

CO5 - Builds procedural skills in routing processes and IP addressing implementation.

CO6 - Supports procedural understanding of reliable data transfer using transport layer protocols.

CO7 - Provides moderate procedural knowledge for managing security and congestion control mechanisms.

3. Justification of PO3 to ALL COs:

CO1 - Provides foundational knowledge supporting basic logical understanding of networking systems.

CO2 - Develops logical thinking through layered communication and protocol interaction analysis.

CO3 - Enhances analytical and problem-solving ability in selecting suitable transmission methods and topologies.

CO4 - Strengthens logical reasoning for detecting and correcting transmission errors.

CO5 - Develops strong problem-solving skills in routing decisions and IP addressing mechanisms.

CO6 - Supports logical analysis of reliable data transmission and flow control mechanisms.

CO7 - Improves problem-solving capability in handling network security threats and congestion issues.

4. Justification of PO4 to ALL COs:

CO1 - Enables understanding of networking terminology useful for basic technical communication.

CO2 - Helps students explain networking models using standard technical communication practices.

CO3 - Develops ability to communicate analysis of transmission techniques and network structures.

CO4 - Supports explanation of protocol operations and error control mechanisms in technical discussions.

CO5 - Enhances professional communication while describing routing processes and IP addressing schemes.

CO6 - Improves technical communication related to transport layer services and data transfer mechanisms.

CO7 - Strengthens professional communication skills in discussing network security and performance issues.

5. Justification of PO3 to ALL COs:

CO1 - Provides basic knowledge required for initial analytical understanding of networking systems.

CO2 - Supports analytical judgment in selecting appropriate network models for communication processes.

CO3 - Develops strong decision-making ability in choosing suitable transmission media and network topologies.

CO4 - Enhances analytical skills for identifying errors and selecting appropriate correction techniques.

CO5 - Builds decision-making capability in routing selection and IP addressing strategies.

CO6 Supports analysis of transport protocols for efficient and reliable data communication.

CO7 - Strengthens analytical judgment in handling network security threats and congestion management.

6. Justification of PO4 to ALL COs:

CO1 - Provides foundational networking knowledge supporting employability in IT-related fields.

CO2 - Develops understanding of network models useful for designing innovative communication solutions.

CO3 - Enhances creative thinking in designing efficient network structures and communication systems.

CO4 - Supports skill development required for implementing reliable networking solutions in industry.

CO5 Builds employability skills through practical understanding of routing and network configuration.

CO6 - Strengthens professional competency in managing transport layer communication services.

CO7 - Encourages entrepreneurial and industry-oriented thinking in network security and performance management.

7. Justification of PO3 to ALL COs:

CO1 - Provides basic networking knowledge applicable across multiple computing and communication domains.

CO2 - Develops interdisciplinary understanding of standardized communication models used in various technologies.

CO3 - Enhances competence in integrating networking concepts with telecommunications and distributed systems.

CO4 - Supports interdisciplinary application of data communication and error control mechanisms.

CO5 - Builds capability to apply networking concepts in cloud computing and internet-based applications.

CO6 Enables understanding of transport protocols used across diverse software and communication platforms.

CO7 - Promotes interdisciplinary awareness through integration of cybersecurity, networking, and system performance concepts.

8. Justification of PO4 to ALL COs:

CO1 - Provides awareness of responsible use of networking technologies in society.

CO2 - Supports understanding of standardized communication practices promoting ethical technology usage.

CO3 - Encourages responsible selection of communication systems considering societal connectivity needs.

CO4 - Develops ethical understanding in maintaining reliable and error-free data communication.

CO5 - Promotes responsible network management ensuring fair and secure data transmission.

CO6 - Supports ethical handling of data communication services across network platforms.

CO7 - Strongly relates to ethical practices through network security, privacy protection, and safe information sharing.

9. Justification of PO3 to ALL COs:

CO1 - Provides foundational networking knowledge applicable to local and indigenous communication systems.

CO2 - Supports understanding of globally accepted models adaptable to regional technological practices.

CO3 - Enables application of networking solutions suitable for local infrastructure and communication needs.

CO4 - Promotes reliable communication practices adaptable to indigenous technological environments.

CO5 - Helps integrate modern routing and addressing techniques in regional and community networks.

CO6 - Provides basic support for implementing contemporary transport protocols in diverse environments.

CO7 - securing digital infrastructure while supporting safe and inclusive technological adoption.

10. Justification of PO4 to ALL COs:

CO1 - Provides basic knowledge required for understanding network system planning concepts.

CO2 - Supports structured planning of communication systems using layered network models.

CO3 - Develops strong capability in designing network solutions based on transmission methods and topologies.

CO4 - Enhances solution development skills through implementation of reliable data link protocols.

CO5 - Builds competency in planning efficient routing and addressing schemes for network systems.

CO6 - Supports development of communication solutions using transport layer services.

CO7- Strengthens system development ability by addressing network security and performance challenges.

11. Justification of PO3 to ALL COs:

CO1 - Creates awareness about responsible use of networking infrastructure in society.

CO2 - Promotes ethical understanding of standardized communication practices.

CO3 - Encourages socially responsible implementation of communication technologies.

CO4 - Supports ethical responsibility in maintaining accurate and reliable data transmission.

CO5 - Develops accountability in managing network resources and communication systems.-

CO6 - Enhances responsible handling of end-to-end communication services.

CO7- Ethical awareness through cyber security, privacy protection, and safe network usage.

12. Justification of PO4 to ALL COs:

CO1 - Provides fundamental knowledge supporting basic research understanding in networking domains.

CO2 - Develops methodological understanding of communication models used in networking research.

CO3 - Enhances analytical and investigative skills for studying transmission techniques and network

performance.

CO4 - Supports research competence through analysis of protocols and error control mechanisms.

CO5 - Builds methodological skills in evaluating routing algorithms and addressing schemes.

CO6 - Enables study and comparison of transport protocols for research-based communication solutions.

CO7- Strengthens research capability in network security analysis and performance optimization.

13. Justification of PO3 to ALL COs:

CO1 - Provides basic networking knowledge required for participation in team-based technical activities.

CO2 - Supports collaborative understanding of layered network communication in group learning environments.

CO3 - Enhances teamwork skills while analyzing and designing network structures collaboratively.

CO4 - Encourages cooperative implementation of protocols and error handling mechanisms.

CO5 - Develops team-based problem-solving skills in routing and network configuration tasks.

CO6 - Supports collaborative management of communication services in networking projects.

CO7 - Promotes teamwork in addressing network security challenges and performance management.

14. Justification of PO4 to ALL COs:

CO1 - Provides essential networking fundamentals required for applying specialized domain knowledge.

CO2 - Supports application of standardized networking models in specialized communication systems.

CO3 - Applies specialized knowledge in designing and analyzing network transmission systems.

CO4 - Enhances application of specialized skills in implementing data link protocols and error control techniques.

CO5 - Develops expertise in applying routing algorithms and IP addressing in real-world networks.

CO6 - Supports specialized application of transport layer services for efficient communication systems.

CO7 - Strengthens specialized knowledge application in network security and performance optimization.

15. Justification of PO3 to ALL COs:

CO1 - Provides awareness of networking infrastructure and its environmental impact.

CO2 - Supports understanding of efficient communication models contributing to resource optimization.

CO3 - Encourages selection of transmission media and topologies considering energy efficiency and risk factors.

CO4 - Promotes reliable data communication reducing operational risks in network systems.

CO5 - Supports risk management through efficient routing and controlled network traffic handling.

CO6 Helps manage communication risks using reliable transport layer protocols.

CO7- Strongly relates to risk management through network security, threat prevention, and safe system operation.

**CBCS Syllabus as per NEP 2020 for T.Y.B.C.A (Science)
(2024 Pattern)**

Name of the Programme	: B.C.A. (Science)
Programme Code	: BCA
Class	: T.Y.B.C.A.
Semester	: V
Course Type	: Major Mandatory [Theory]
Course Code	: BCA-304-MRM
Course Title	: Software Engineering
No. of Credits	: 02
No. of Teaching Hours	: 30

Course Objectives:

1. To understand the basic concepts of systems and software engineering.
2. To learn the need and importance of software engineering.
3. To study the phases of the Software Development Life Cycle (SDLC).
4. To understand different software process models and agile methodology.
5. To learn requirement gathering and analysis techniques.
6. To develop skills in preparing basic software documentation like SRS.
7. To apply structured analysis techniques such as DFD and Data Dictionary.

Course Outcomes:

By the end of the course, students will be able to:

- CO1:** Understand the basic concepts of systems and software engineering.
- CO2:** Explain the need and importance of software engineering.
- CO3:** Describe the phases of the Software Development Life Cycle (SDLC).
- CO4:** Compare different software process models and Agile methodology.
- CO5:** Apply requirement gathering and analysis techniques.
- CO6:** Prepare basic software documentation such as the Software Requirements Specification (SRS).
- CO7:** Apply structured analysis techniques such as Data Flow Diagrams (DFD) and Data Dictionary.

Topics and Learning Points

Units	Title & Content	No. of Lectures
Unit 1	<p>UNIT I: Fundamentals of Software Engineering</p> <p>1.1 Introduction to System</p> <ul style="list-style-type: none"> • Definition of System • Elements of a System (Input, Output, Process, Feedback, Boundary) • Types of Systems (Open/Closed, Physical/Abstract, Computer-based systems) <p>1.2 Introduction to Software</p> <ul style="list-style-type: none"> • Definition and Characteristics of Software • Software Application Areas • Need for Software Engineering <p>1.3 Software Engineering Basics</p> <ul style="list-style-type: none"> • Definition and Objectives • Software Crisis • Role of Software Engineer • Good Software Characteristics 	06 L
Unit 2	<p>UNIT II: Software Development Life Cycle (SDLC)</p> <p>2.1 Introduction to SDLC</p> <ul style="list-style-type: none"> • Meaning and Importance • Phases of SDLC <p>2.2 Phases of SDLC</p> <ul style="list-style-type: none"> • Feasibility Study (Technical, Economic, Operational) • Requirement Analysis • System Design (Overview only) • Coding • Testing <ul style="list-style-type: none"> ○ Unit Testing ○ Integration Testing ○ System Testing • Implementation • Maintenance <p>2.3 Software Documentation</p> <ul style="list-style-type: none"> • Importance of Documentation 	08 L

	<ul style="list-style-type: none"> Types (SRS, Design Document, User Manual – overview) 	
Unit 3	<p>UNIT III: Software Process Models & Agile</p> <p>3.1 Introduction to Process Models</p> <p>3.2 Traditional Models</p> <ul style="list-style-type: none"> Waterfall Model Incremental Model Prototyping Model Spiral Model (Basic Concept only) <p>3.3 Agile Development</p> <ul style="list-style-type: none"> Introduction to Agile Agile Principles Agile vs Traditional Models Extreme Programming (XP) – basic practices 	07 L
Unit 4	<p>UNIT IV: Requirements Engineering & Structured Analysis</p> <p>4.1 Requirements Engineering</p> <ul style="list-style-type: none"> Introduction Types of Requirements (Functional & Non-functional) Requirement Engineering Process SRS (Structure & Components) <p>4.2 Requirement Gathering Techniques</p> <ul style="list-style-type: none"> Interview Questionnaire Observation Record Review <p>4.3 Structured Analysis</p> <ul style="list-style-type: none"> Data Flow Diagram (DFD) <ul style="list-style-type: none"> Symbols Context Diagram Level 1 DFD Data Dictionary (Basic Concepts) 	09 L

Reference /Text Books:

1. Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
2. Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.
3. Awad, E. M. (2004). *Systems analysis and design* (2nd ed.). Galgotia Publications.
4. Senn, J. A. (1989). *Analysis and design of information systems* (2nd ed.). McGraw-Hill.

5. Jalote, P. (2005). *An integrated approach to software engineering* (3rd ed.). Springer.
6. Aggarwal, K. K., & Singh, Y. (2007). *Software engineering* (3rd ed.). New Age International Publishers.
7. Beck, K. (2000). *Extreme programming explained: Embrace change*. Addison-Wesley.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO1 12	PO 13	PO 14	PO 15
CO1	3	2	1	1	1	1	2	1	1	2	1	1	1	2	1
CO2	3	2	2	1	2	2	2	1	1	2	2	1	1	2	1
CO3	2	3	2	1	2	2	1	1	1	3	2	2	2	2	1
CO4	2	2	3	2	3	3	2	2	1	3	2	2	3	3	2
CO5	2	3	3	2	3	3	2	2	2	3	3	3	3	3	2
CO6	1	3	2	3	2	2	1	2	1	3	3	2	2	3	1
CO7	1	3	3	2	3	2	2	1	1	3	2	2	2	3	1

Weight: 1 - Partiallyrelated 2 - Moderately Related 3 - Strongly Related

Justification of Mapping of PO1 with All CO'S

CO1: PO1: Understanding system and software engineering fundamentals builds strong interdisciplinary and integrated knowledge.

CO2:PO1: Explaining the importance of software engineering strengthens broad professional and applied knowledge.

CO3:PO1: SDLC phases provide structured understanding contributing to integrated learning.

CO4:PO1: Comparison of models enhances awareness of diverse engineering approaches.

CO5:PO1: Requirement analysis applies foundational knowledge across domains.

CO6:PO1: Documentation supports conceptual understanding at a basic level.

CO7:PO1: Structured analysis techniques partially contribute to integrated knowledge.

Justification of Mapping of PO2 with All CO'S

CO1:PO2: Conceptual clarity supports procedural learning.

CO2:PO2: Importance of SE encourages systematic skill development.

CO3:PO2: SDLC phases directly reflect procedural understanding.

CO4:PO2: Process model comparison enhances methodological insight.

CO5:PO2: Requirement analysis demands strong procedural application.

CO6:PO2: SRS preparation is a structured procedural task.

CO7:PO2: DFDs and Data Dictionary are core procedural tools.

Justification of Mapping of PO3 with All CO'S

CO1:PO3: Basic concepts introduce problem-identification skills.

CO2:PO3: Understanding SE importance aids analytical reasoning.

CO3:PO3: SDLC phases support logical sequencing of solutions.

- CO4:PO3:** Model comparison enhances critical and logical thinking.
CO5:PO3: Requirement analysis involves strong problem-solving.
CO6:PO3: Documentation supports analytical clarity.
CO7:PO3: Structured analysis strongly develops problem-solving ability.
-

Justification of Mapping of PO4 with All CO'S

- CO1:PO4:** Concept understanding supports basic technical communication.
CO2:PO4: Explaining SE improves verbal articulation.
CO3:PO4: SDLC descriptions enhance written expression.
CO4:PO4: Model comparison improves technical discussion skills.
CO5:PO4: Requirement gathering involves stakeholder communication.
CO6:PO4: SRS preparation requires strong professional documentation skills.
CO7:PO4: Diagrammatic representation enhances visual communication.
-

Justification of Mapping of PO5 with All CO'S

- CO1:PO5:** Foundational concepts support analytical judgment.
CO2:PO5: Understanding importance aids informed decision-making.
CO3:PO5: SDLC analysis supports evaluative thinking.
CO4:PO5: Model comparison develops strong analytical judgment.
CO5:PO5: Requirement analysis requires decision-making skills.
CO6:PO5: Documentation reflects analytical precision.
CO7:PO5: Structured techniques enable systematic analysis.
-

Justification of Mapping of PO6 with All CO'S

- CO1:PO6:** Basic SE concepts support employability readiness.
CO2:PO6: SE importance aligns with professional preparedness.
CO3:PO6: SDLC knowledge enhances industry relevance.
CO4:PO6: Agile methodology encourages adaptability and creativity.
CO5:PO6: Requirement analysis develops practical employability skills.
CO6:PO6: Documentation supports professional practice.
CO7:PO6: Analysis tools prepare students for real-world tasks.
-

Justification of Mapping of PO7 with All CO'S

- CO1:PO7:** Systems concepts promote interdisciplinary understanding.
CO2:PO7: SE relevance spans multiple domains.
CO3:PO7: SDLC offers limited interdisciplinary exposure.
CO4:PO7: Agile promotes collaborative and interdisciplinary thinking.
CO5:PO7: Requirement gathering integrates multiple perspectives.
CO6:PO7: Documentation supports limited interdisciplinary linkage.
CO7:PO7: DFDs integrate business and technical viewpoints.
-

Justification of Mapping of PO8 with All CO'S

- CO1:PO8:** Basic system concepts introduce ethical awareness indirectly.
CO2:PO8: Understanding software importance promotes responsible system usage.
CO3:PO8: SDLC phases expose ethical considerations in development stages.
CO4:PO8: Agile methodology emphasizes collaboration and ethical teamwork.
CO5:PO8: Requirement analysis considers stakeholder needs and social responsibility.
CO6:PO8: SRS preparation enforces accountability and ethical documentation.
CO7:PO8: Structured analysis supports ethical clarity at a basic level.
-

Justification of Mapping of PO9 with All CO'S

- CO1:PO9:** Foundational concepts allow minimal integration of traditional practices.

CO2:PO9: Importance of SE may reference contextual or local system needs.

CO3:PO9: SDLC permits limited adaptation to indigenous workflows.

CO4:PO9: Process models can be adapted to local development contexts.

CO5:PO9: Requirement gathering incorporates local and indigenous user needs.

CO6:PO9: Documentation may reflect culturally contextual requirements.

CO7:PO9: Analysis tools marginally support indigenous system representation.

Justification of Mapping of PO10 with All CO'S

CO1:PO10: System fundamentals support solution conceptualization.

CO2:PO10: Understanding SE importance aids system planning awareness.

CO3:PO10: SDLC phases directly support structured system development.

CO4:PO10: Process models and Agile strongly enable solution design.

CO5:PO10: Requirement analysis is critical for solution formulation.

CO6:PO10: SRS preparation is essential for system implementation.

CO7:PO10: DFDs and Data Dictionary strongly support system design.

Justification of Mapping of PO11 with All CO'S

CO1:PO11: Introductory concepts promote basic professional ethics.

CO2:PO11: SE importance highlights responsibility in system development.

CO3:PO11: SDLC emphasizes accountability at each development stage.

CO4:PO11: Agile practices stress ethical collaboration and transparency.

CO5:PO11: Requirement analysis demands ethical stakeholder handling.

CO6:PO11: SRS enforces accountability and professional standards.

CO7:PO11: Structured analysis ensures responsible system representation.

Justification of Mapping of PO12 with All CO'S

CO1:PO12: Basic concepts support introductory research understanding.

CO2:PO12: Understanding SE importance aids exploratory learning.

CO3:PO12: SDLC phases reflect systematic investigation processes.

CO4:PO12: Model comparison develops evaluative research skills.

CO5:PO12: Requirement analysis involves data collection and interpretation.

CO6:PO12: Documentation reflects structured research reporting.

CO7:PO12: DFDs support analytical representation of research findings.

Justification of Mapping of PO13 with All CO'S

CO1:PO13: Concepts introduce collaborative development awareness.

CO2:PO13: SE importance emphasizes teamwork in projects.

CO3:PO13: SDLC promotes role-based team coordination.

CO4:PO13: Agile methodology strongly emphasizes team collaboration.

CO5:PO13: Requirement gathering requires stakeholder and team interaction.

CO6:PO13: Documentation supports coordinated team efforts.

CO7:PO13: Structured analysis tools support collaborative design.

Justification of Mapping of PO14 with All CO'S

CO1:PO14: System concepts build domain-specific understanding.

CO2:PO14: SE importance supports application-oriented learning.

CO3:PO14: SDLC applies theoretical knowledge in practice.

CO4:PO14: Process models demonstrate applied software expertise.

CO5:PO14: Requirement analysis applies specialized SE techniques.

CO6:PO14: SRS preparation demonstrates in-depth subject application.

CO7:PO14: DFDs and Data Dictionary reflect specialized system knowledge.

Justification of Mapping of PO15 with All CO'S

CO1:PO15: Basic system knowledge supports awareness of environmental impact.

CO2:PO15: SE importance highlights sustainable system practices.

CO3:PO15: SDLC allows limited consideration of environmental risks.

CO4:PO15: Agile supports resource-efficient development practices.

CO5:PO15: Requirement analysis may include environmental constraints.

CO6:PO15: Documentation supports compliance with environmental standards.

CO7:PO15: Structured analysis marginally supports environmental planning.

**CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science)
(2024 Pattern)**

Name of the Programme : B.C.A. (Science)

Program Code : BCA
Class : T.Y.B.C.A
Semester : V
Course Type : Major Mandatory [Practical]
Course Name : Lab Course On BCA-301
Course Code : BCA-305-MRM
No. of Credits : 02
No. of Practical Hours : 60

Course Objectives:

1. To strengthen understanding of object-oriented programming concepts using Java.
2. To develop skills in inheritance, interfaces, and polymorphism for building reusable and modular applications.
3. To gain practical experience with collections, exception handling, and I/O operations for robust programming.
4. To design GUI applications using Swing for interactive user interfaces.
5. To integrate Java applications with databases using JDBC for persistent data management.
6. To explore web application development using Servlets and JSP for dynamic, server-side programming.

Course Outcomes:

At the end of the course, students will be able to:

CO1: Apply Java Collections Framework (List, Set, Map) to efficiently manage and manipulate data in applications.

CO2: Develop multithreaded Java applications using thread life cycle, synchronization, and inter-thread communication.

CO3: Design GUI-based applications using Swing components and layout managers.

CO4: Implement event handling mechanisms including mouse, keyboard, and action events.

CO5: Develop database-driven applications using JDBC with transaction management and metadata handling.

CO6: Integrate multithreading, GUI, collections, and database connectivity to build complete Java-based systems.

CO7: Demonstrate professional ethics, optimized resource utilization, and secure programming practices in Java applications.

Topics and Learning Points

Sr. No.	Assignment Name	No. of Assignments
1	(Collections – List Implementation)	2
2	(Collections – Set Implementation)	2
3	(Collections – Map & Interfaces)	2
4	(Comparator & Enumeration)	2
5	(Multithreading – Basics)	2
6	(Multithreading – Advanced)	2
7	(Swing – Basic Components)	2
8	(Swing – Advanced Components & Dialogs)	2
9	(Event Handling)	2
10	(JDBC – Basic Connectivity)	2
11	(JDBC – Advanced Concepts)	2
12	(Mini Project Integration)	2

Reference /Text Books:

Text Books:

1. **Advanced Java Programming** by Uttam Kumar Roy (Oxford University Press, 2015) – A comprehensive book covering advanced concepts like JDBC, Servlets, JSP, and Swing.
2. **Advanced Java Programming** by A. A. Puntambekar (Technical Publications, 2021) – Tailored for academic courses, with clear explanations and lab exercises.
3. **Advanced Java Programming** by A.S. Arunachalam & S. Perumal (2019) – Focuses on practical aspects of advanced Java topics.
4. **Core Java: An Integrated Approach** by R. Nageswara Rao – Covers both fundamentals and advanced features up to Java 8, useful for bridging basics with advanced topics.

Reference Books:

1. **Java: The Complete Reference** by Herbert Schildt – A classic, detailed reference for both core and advanced Java concepts.
2. **Head First Servlets and JSP** by Bryan Basham, Kathy Sierra, and Bert Bates – Excellent for understanding web technologies in Java with practical examples.
3. **Java Programming with Oracle JDBC** by Donald Bales – Focused on database programming with JDBC.
4. **Effective Java** by Joshua Bloch – Provides best practices and design patterns for writing robust Java code.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PO13	PO14	PO15
CO1	3	3	3	3	3	3	3	2	2	3	2	3	3	3	3
CO2	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3
CO3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2
CO4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
CO5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
CO6	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
CO7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S**PO1: Broad Knowledge and Integrated Learning**

CO1: Java Collections Framework strengthens fundamental programming knowledge and enhances structured data handling capability.

CO2: Multithreading concepts integrate operating system principles with Java programming, expanding interdisciplinary understanding.

CO3: GUI development using Swing integrates programming logic with user-interface design concepts.

CO4: Event handling connects system-level interaction with application-level logic, improving integrated technical understanding.

CO5: JDBC connects Java applications with database systems, broadening knowledge across backend technologies.

CO6: Integration of collections, GUI, multithreading, and database builds comprehensive system-level development skills.

CO7: Ethical programming and optimized coding practices promote holistic and responsible computing knowledge.

Mapping of PO2 with All CO'S

PO2: Procedural Understanding for Skill Building

CO1: Implementing List, Set, and Map develops structured programming and procedural coding skills.

CO2: Creating and managing threads enhances systematic execution control and logical implementation ability.

CO3: Designing GUI layouts builds step-by-step application development skills.

CO4: Event handling strengthens practical implementation of interactive programming techniques.

CO5: JDBC connectivity improves structured database interaction skills.

CO6: Integrating multiple technologies enhances advanced procedural implementation capability.

CO7: Applying secure and optimized coding practices strengthens professional technical competence.

Mapping of PO3 with All CO'S

PO3: Logical Thinking and Problem-Solving Capability

CO1: Selecting appropriate data structures improves logical analysis and decision-making ability.

CO2: Managing thread synchronization develops critical problem-solving skills.

CO3: GUI component arrangement improves analytical thinking in interface design.

CO4: Handling runtime events enhances logical debugging capability.

CO5: SQL query execution strengthens data-oriented problem-solving ability.

CO6: Integrated system development improves comprehensive solution-building skills.

CO7: Ethical and optimized system design enhances responsible problem-solving approaches.

Mapping of PO4 with All CO'S

PO4: Professional and Technical Communication Skills

CO1: Organized collection usage improves clarity and structured code presentation.

CO2: Multithreading documentation enhances technical communication.

CO3: GUI design improves user-system interaction communication.

CO4: Event-driven programming enhances interactive communication between user and system.

CO5: Database reporting improves structured information communication.

CO6: Integrated applications enhance overall system documentation clarity.

CO7: Professional coding standards promote effective technical communication practices.

Mapping of PO5 with All CO'S

PO5: Analytical Judgment and Decision-Making Skills

CO1: Choosing suitable data structures improves analytical evaluation skills.

CO2: Managing thread priorities strengthens performance-based decision-making ability.

CO3: Selecting appropriate layouts enhances design evaluation skills.

CO4: Event listener implementation improves technical judgment.

CO5: Database transaction management strengthens logical decision-making capability.

CO6: Integrating technologies requires high-level system evaluation.

CO7: Secure programming practices enhance risk-based analytical judgment.

Mapping of PO6 with All CO'S

PO6: Creativity, Employability, and Entrepreneurial Mindset

CO1: Collections are widely used in enterprise-level applications, enhancing employability.

CO2: Multithreading improves performance-oriented professional development.

CO3: GUI development enhances desktop application development skills.

CO4: Event-driven applications strengthen real-world project capability.

CO5: JDBC knowledge supports backend and enterprise application development.

CO6: Full system integration fosters innovative and entrepreneurial thinking.

CO7: Secure and optimized application design supports industry readiness.

Mapping of PO7 with All CO'S

PO7: Interdisciplinary Awareness and Competence

CO1: Collections manage data across business, scientific, and management domains.

CO2: Multithreading supports high-performance systems in various industries.

CO3: GUI applications apply to banking, education, and logistics systems.

CO4: Event handling connects software with real-world user interaction.

CO5: Database systems integrate business information systems.

CO6: Integrated Java systems support enterprise-level interdisciplinary solutions.

CO7: Ethical programming ensures responsible use across domains.

Mapping of PO8 with All CO'S

PO8: Ethical Values through Social Participation

CO1: Efficient data management promotes responsible resource utilization.

CO2: Controlled thread execution prevents system misuse.

CO3: User-friendly GUI ensures inclusive and accessible systems.

CO4: Proper event handling prevents application misuse.

CO5: Secure database connectivity ensures data privacy and integrity.

CO6: Integrated secure systems promote ethical digital participation.

CO7: Professional ethics in coding supports responsible computing practices.

Mapping of PO9 with All CO'S

PO9: Integration of Indigenous Knowledge with Contemporary Practice

CO1: Traditional data organization is digitized using collections.

CO2: Multithreading enhances efficiency of traditional workflows in digital form.

CO3: GUI systems modernize manual processes.

CO4: Event-based interaction digitizes conventional system responses.

CO5: Databases transform traditional record-keeping into digital systems.

CO6: Integrated Java applications modernize conventional service models.

CO7: Ethical system design ensures responsible digital transformation.

Mapping of PO10 with All CO'S

PO10: System Planning and Solution Development

CO1: Collections form the backend data structure planning of systems.

CO2: Multithreading supports scalable system design.

CO3: GUI design forms user-interface planning.

CO4: Event handling ensures interactive solution development.

CO5: JDBC manages system-level data storage solutions.

CO6: Integration of technologies enables complete system development.

CO7: Secure system architecture supports sustainable solution planning.

Mapping of PO11 with All CO'S

PO11: Ethical Awareness and Social Accountability

CO1: Clean coding practices promote responsible development.

CO2: Proper synchronization prevents data corruption.

CO3: GUI accessibility ensures inclusive software development.

CO4: Event validation prevents misuse.

CO5: Secure database transactions ensure accountability.

CO6: Integrated applications emphasize responsible system development.

CO7: Ethical standards promote socially accountable computing.

Mapping of PO12 with All CO'S

PO12: Research Competence and Methodological Skills

CO1: Collections support algorithm experimentation and research analysis.

CO2: Multithreading improves performance testing experiments.

CO3: GUI prototypes support research-based application development.

CO4: Event-driven models assist interactive system experimentation.

CO5: JDBC supports data-driven research systems.

CO6: Integrated Java systems enhance applied research methodology.

CO7: Secure and optimized coding improves research reliability.

Mapping of PO13 with All CO'S

PO13: Collaborative and Team-Based Skills

CO1: Collection-based tasks promote collaborative coding.

CO2: Thread-based applications require coordinated debugging.

CO3: GUI modules can be divided among team members.

CO4: Event handling improves coordinated module development.

CO5: Database applications involve backend team collaboration.

CO6: Integrated systems strengthen teamwork and coordination skills.

CO7: Professional ethics promote respectful team collaboration.

Mapping of PO14 with All CO'S

PO14: Specialized Knowledge Application

CO1: Advanced collection usage strengthens Java specialization.

CO2: Multithreading enhances advanced concurrency knowledge.

CO3: GUI development builds subject-specific expertise.

CO4: Event-driven programming strengthens application specialization.

CO5: JDBC develops backend specialization skills.

CO6: Integration of multiple technologies builds comprehensive Java mastery.

CO7: Secure system development enhances professional expertise.

Mapping of PO15 with All CO'S

PO15: Environmental Responsibility and Risk Management

CO1: Efficient data structures reduce memory and resource wastage.

CO2: Controlled multithreading reduces system overload risks.

CO3: Optimized GUI design minimizes unnecessary resource consumption.

CO4: Proper event validation prevents runtime errors and system risks.

CO5: Transaction management prevents data loss and operational risks.

CO6: Integrated system design ensures sustainable software development.

CO7: Secure and optimized applications reduce cybersecurity risks and promote environmental sustainability.

**CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science)
(2024 Pattern)**

Name of the Programme	: B.C.A. (Science)
Program Code	: BCA
Class	: T.Y.B.C.A
Semester	: V
Course Type	: Major Mandatory [Practical]
Course Name	: Lab Course on BCA-302
Course Code	: BCA-306-MRM
No. of Credits	: 02
No. of Practical Hours	: 60

Prerequisites:

Understanding of Problem solving techniques using a programming language and basic data structures.

Course Objectives:

1. Develop modular Python programs.
2. Apply suitable Python programming constructs, built-in data structures using Python libraries to solve a problem.
3. Implement functions and modules to enhance code reusability and maintainability
4. Demonstrate proficiency in handling different data structures such as lists, tuples, and dictionaries in Python.
5. Understand basic File handling in Python.
6. Apply exception handling mechanisms to build robust and error-free Python programs.
7. Utilize Python libraries for data visualization and mathematical computations.

Course Outcomes:

At the end of the course, students will be able to:

- CO1:** Students will be able to develop modular Python programs.
- CO2:** Students will be able to Apply suitable Python programming constructs, built-in data structures using Python libraries to solve a problem.
- CO3:** Students will be able to implement functions and modules to enhance code reusability and maintainability.
- CO4:** Students will be able to demonstrate proficiency in handling different data structures such as lists, tuples, and dictionaries in Python.

CO5: Students will be able to understand basic File handling in Python.

CO6: Students will be able to apply exception handling mechanisms to build robust and error-free Python programs.

CO7: Students will be able to utilize Python libraries for data visualization and mathematical computations.

Topics and Learning Points

Sr. No.	Assignment Name
Assignment 1	Practical implementation of Getting Started with Python (Hello World, variables, input/output, type conversion)
Assignment 2	Practical implementation of Basic Operators and Expressions (Arithmetic, relational, logical, bitwise operations)
Assignment 3	Practical implementation of Conditional Statements in Python (if, if-else, nested if, simple decision making)
Assignment 4	Practical implementation of Looping Constructs (for, while, nested loops, break, continue, pass)
Assignment 5	Practical implementation of functions in Python (built-in functions, user-defined functions, recursion, lambda)
Assignment 6	Practical implementation of Working with Modules and Packages (importing modules, creating user-defined modules)
Assignment 7	Practical implementation of Lists and Tuples (basic operations, indexing, slicing, simple applications)
Assignment 8	Practical implementation of Sets and Dictionaries (set operations, dictionary key-value pairs, simple programs)
Assignment 9	Practical implementation of File Handling in Python (text files, CSV, JSON, Pickle basics)
Assignment 10	Practical implementation of Exception Handling (try-except, finally, raise, handling common errors)
Assignment 11	Practical implementation of Python libraries for data visualization (NumPy, Pandas, Seaborn)
Assignment 12	Practical implementation of Python libraries for mathematical computations. (matplotlib, math)

Reference /Text Books:

Text Books:

1. Venkatesh, Nagaraju Y, Introduction to Python Programming, Khanna

- Publishing House, 2021.
- Jeeva Jose, Introduction to Computing & Problem Solving With PYTHON, Khanna Publishing House, 2023.
 - Sheetal Taneja & Naveen kumar: Python Programming a Modular approach – A Modular approach with Graphics, Database, Mobile and Web applications, Pearson, 2017.

Reference Books:

- Think Python, by Allen Downey, 2nd edition, 2015, O'Reilly. https://drive.google.com/file/d/1p9Pul6d5UvnOrO9-QLE2_p4YvMk5cIg/view
- An introduction to Python for absolute beginners, by Bob Dowling, Cambridge Univ.
- Introduction to Computation and Programming using Python, by John Guttag, 2nd edition, 2016, PHI India.

Web Resources:

- <https://www.learnpython.org/>
- <https://www.w3schools.com/python/default.asp>

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PO13	PO14	PO15
CO1	3	3	3	3	3	2	2	3	3	2	2	3	2	3	2
CO2	3	3	3	3	3	2	2	3	3	2	2	3	2	3	2
CO3	3	3	3	3	3	3	2	3	3	2	2	3	2	3	2
CO4	3	3	2	3	3	2	2	2	3	1	1	2	1	2	1
CO5	2	2	2	2	2	1	3	2	3	1	2	2	2	2	2
CO6	3	3	2	3	3	2	2	3	2	2	3	3	3	3	2
CO7	2	2	3	2	3	3	3	3	3	3	3	3	3	3	2

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

Mapping of PO1 (Disciplinary Knowledge) with All COs

CO1: Practical implementation of modular Python programs strengthens fundamental programming knowledge.

CO2: Writing programs using Python constructs enhances applied computational knowledge.

CO3: Implementing functions and modules demonstrates understanding of software design principles.

CO4: Handling lists, tuples, sets, and dictionaries reflects strong data structure knowledge.

CO5: Performing file handling operations builds understanding of input-output systems.

CO6: Applying exception handling improves knowledge of error control mechanisms.

CO7: Using libraries like NumPy/Matplotlib strengthens applied mathematical and computational knowledge.

Mapping of PO2 (Critical Thinking & Problem Solving)

CO1: Modular programming improves structured problem-solving ability.

CO2: Using conditional and looping constructs enhances analytical thinking.

CO3: Designing reusable functions supports logical breakdown of problems.

CO4: Data structure manipulation improves computational reasoning.

CO5: File operations develop logical data processing skills.

CO6: Exception handling strengthens debugging and troubleshooting ability.

CO7: Visualization tasks improve interpretation and analytical skills.

Mapping of PO3 (Innovation & Entrepreneurship)

CO1: Modular coding supports scalable software development.

CO2: Python constructs help build innovative software solutions.

CO3: Reusable modules are essential for startup and product development.

CO4: Efficient data handling enables optimized application design.

CO5: File management supports business data applications.

CO6: Robust error handling ensures product reliability.

CO7: Data visualization enables data-driven entrepreneurial decisions.

Mapping of PO4 (Research & Analytical Skills)

CO1: Structured programs assist in developing research tools.

CO2: Logical constructs help implement research algorithms.

CO3: Modularization improves experimentation and testing.

CO4: Data structures help manage research datasets efficiently.

CO5: File handling supports data storage and retrieval in research.

CO6: Exception handling ensures data accuracy and reliability.

CO7: Visualization aids in presenting research findings.

Mapping of PO5 (Technical Competence)

CO1: Practical coding enhances technical proficiency.

CO2: Construct-based programming strengthens implementation skills.

CO3: Functions and modules improve maintainable coding practices.

CO4: Data structures enhance computational efficiency.

CO5: File handling builds real-world application skills.

CO6: Error handling improves robustness of systems.

CO7: Library usage enhances advanced technical capability.

Mapping of PO6 (Communication & Teamwork)

CO1: Modular programming improves code readability in teams.

CO2: Logical coding structure enhances collaborative development.

CO3: Reusable functions support teamwork in large projects.

CO4: Data structure usage aids in technical discussions.

CO5: File handling assists in data sharing within teams.

CO6: Exception messages improve clarity in debugging.

CO7: Visualization improves presentation and communication of results.

Mapping of PO7 (Research & Development Skills)

CO1: Structured coding supports development of research prototypes.

CO2: Python constructs assist in algorithm implementation.

- CO3: Modules support scalable research applications.
 - CO4: Data structures enable efficient research data handling.
 - CO5: File operations are essential for storing research data.
 - CO6: Exception handling ensures reliability in research programs.
 - CO7: Data visualization supports analytical research interpretation.
-

Mapping of PO8 (Lifelong Learning)

- CO1: Modular coding promotes self-learning practices.
 - CO2: Logical constructs encourage independent coding improvement.
 - CO3: Learning modules builds adaptability to new technologies.
 - CO4: Data structure mastery enhances continuous skill development.
 - CO5: File handling introduces real-world data concepts.
 - CO6: Debugging practices encourage continuous improvement.
 - CO7: Library exploration promotes ongoing learning.
-

Mapping of PO9 (Digital Proficiency)

- CO1: Modular programming enhances digital coding skills.
 - CO2: Python constructs improve computational literacy.
 - CO3: Functions improve efficient digital tool usage.
 - CO4: Data structures strengthen data processing skills.
 - CO5: File handling ensures effective digital data management.
 - CO6: Exception handling enhances software reliability.
 - CO7: Visualization libraries enhance ICT competency.
-

Mapping of PO10 (Teamwork & Leadership)

- CO1: Modular coding supports collaborative software projects.
 - CO2: Logical structuring aids inclusive problem-solving.
 - CO3: Functions support shared code development.
 - CO4: Data structures contribute to collaborative applications.
 - CO5: File handling assists in shared data projects.
 - CO6: Reliable applications support team accountability.
 - CO7: Visualization improves team understanding of data insights.
-

Mapping of PO11 (Ethics & Sustainability)

- CO1: Modular programming supports efficient resource use.
 - CO2: Logical coding promotes responsible computing.
 - CO3: Reusable code reduces redundancy and promotes sustainability.
 - CO4: Efficient data structures reduce computational waste.
 - CO5: Proper file management ensures responsible data usage.
 - CO6: Exception handling prevents system failure and data loss.
 - CO7: Visualization supports environmental and social data analysis.
-

Mapping of PO12 (Self-Management & Accountability)

- CO1: Modular coding promotes organized work habits.
 - CO2: Logical implementation builds independent responsibility.
 - CO3: Functions support project management skills.
 - CO4: Data structures improve structured project execution.
 - CO5: File handling requires careful data responsibility.
 - CO6: Error handling shows accountability in coding.
 - CO7: Visualization supports responsible reporting.
-

Mapping of PO13 (Community & Social Responsibility)

CO1: Modular programming supports development of community applications.

CO2: Python constructs help design socially useful software.

CO3: Functions enable scalable community-based solutions.

CO4: Data structures assist in organizing community data.

CO5: File handling supports storage of social datasets.

CO6: Exception handling ensures reliability of public software.

CO7: Visualization aids in presenting community-related insights.

Mapping of PO14 (Environment & Sustainability Awareness)

CO1: Efficient coding reduces computational resource usage.

CO2: Structured programming promotes sustainable development practices.

CO3: Reusable modules reduce development redundancy.

CO4: Optimized data structures reduce processing load.

CO5: File handling promotes efficient digital storage use.

CO6: Robust programs prevent data waste and failures.

CO7: Visualization assists in analyzing environmental datasets.

Mapping of PO15 (Professional & Career Skills)

CO1: Modular programming builds industry-required coding standards.

CO2: Python constructs prepare students for professional roles.

CO3: Functions and modules enhance software development competency.

CO4: Data structures are essential for technical interviews.

CO5: File handling is required in real-world applications.

CO6: Exception handling improves professional code quality.

CO7: Library usage prepares students for analytics and industry tools.

CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science) (2024 Pattern)

Name of the Programme	: B.C.A. (Science)
Program Code	: BCA
Class	: T.Y.B.C.A
Semester	: V
Course Type	: Major Elective (TH)
Course Name	: Operating System
Course Code	: BCA-307-MJE (A)
No. of Credits	: 02
No. of Teaching Hours	: 30

Prerequisites:

- Basic Knowledge of Computer

Course Objectives:

1. To understand Complexity of Operating system as a software. .
2. To understand design issues related to process management and various related algorithms
3. To understand design issues related to memory management and various related algorithms.

Course Outcomes:

At the end of the course, students will be able to:

- CO1:** Understand the working of various operating system modules.
CO2: Explain the concept and algorithms of process scheduling.
CO3: Understand system calls and their role in interacting with the operating system.
CO4: Explain the role of the operating system in memory management.
CO5: Understand process synchronization mechanisms.
CO6: Explain the concept of deadlock and its management techniques.
CO7: Analyze advanced memory management techniques such as paging and segmentation.

Topics and Learning Points

UNIT No.	Chapter Name with Topics	No. of Lectures
----------	--------------------------	-----------------

UNIT-I	<p>Operating System as System Software</p> <p>1.1 What Operating Systems Do – User View, System View, Defining OS</p> <p>1.2 Computer System Architecture – Single processor system, Multiprocessor systems, Clustered Systems</p> <p>1.3 Operating System Operations – Dual mode operation, Timer</p> <p>1.4 Process Management</p> <p>1.5 Memory Management</p> <p>1.6 Operating System Services</p> <p>1.7 User Operating-System Interface – Command interpreter, GUI</p> <p>1.8 System Calls and Types:-Process control, File management, Device management, Information maintenance, Communication, Protection</p>	04
UNIT-II	<p>Process Management and Scheduling</p> <p>2.1 Process Concept – The process, Process states, Process control block.</p> <p>2.2 Process Scheduling – Scheduling queues, Schedulers, context switch</p> <p>2.3 Operations on Process – Process creation with program using fork(), Process termination</p> <p>2.4 Inter-process Communication – Shared memory system, Message passing systems.</p> <p>2.5 Basic Concept – CPU-I/O burst cycle, CPU scheduler, Preemptive scheduling, Dispatcher</p> <p>2.6 Scheduling Criteria</p> <p>2.7 Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round-robin scheduling, Multiple queue scheduling, Multilevel feedback queue scheduling.</p> <p>2.8 Algorithm Evaluation</p>	08
UNIT-III	<p>Process Synchronization and Deadlock</p> <p>3.1 Background</p> <p>3.2 Critical Section Problem</p> <p>3.3 Semaphores: Usage, Implementation</p> <p>3.4 Classic Problems of Synchronization – The bounded buffer problem, The reader writer problem, The dining philosopher problem</p> <p>3.5 System model</p> <p>3.6 Deadlock Characterization – Necessary conditions, Resource allocation graph</p> <p>3.7 Deadlock Prevention</p> <p>3.8 Deadlock Avoidance - Safe state, Resource allocation graph algorithm, Banker's Algorithm</p> <p>3.9 Deadlock Detection</p> <p>3.10 Recovery from Deadlock – Process termination, Resource preemption</p>	08

UNIT- IV	<p>Memory Management</p> <p>4.1. Background – Basic hardware, Address binding, Logical versus physical address space, Dynamic loading, Dynamic linking and shared libraries, Overlays, Swapping</p> <p>4.2 Contiguous Memory Allocation – Memory mapping and protection, Memory allocation, Fragmentation MFT MVT</p> <p>4.3 Paging – Basic Method, Hardware support, Protection, Shared Pages</p> <p>4.4 Segmentation – Basic concept</p> <p>4.5 Demand paging, Performance of demand paging, Page replacement – FIFO, OPT, LRU, MFU, MRU page replacement</p> <p>4.6 Thrashing- Locality Model, Working Set Model, Pre-paging, I/O Interlock,</p>	10
---------------------	--	-----------

Reference /Text Books:

Text Books:

1. **Operating System Concepts** by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne
A widely adopted textbook, currently in its 10th edition, covering fundamental OS principles and modern developments.
2. **Modern Operating Systems** by Andrew S. Tanenbaum and Herbert Bos – A classic text that blends theory with practical examples, especially useful for understanding design and implementation.
3. **Operating Systems: Internals and Design Principles** by William Stallings – Focuses on OS architecture, design issues, and case studies.
4. **Operating Systems: A Concept-Based Approach** by D.M. Dhamdhare – Popular in Indian universities, offering clear explanations and practical exercises.

Reference Books:

1. **The Design of the UNIX Operating System** by Maurice J. Bach – A detailed reference on UNIX internals, useful for advanced learners.
2. **Operating Systems: Principles and Practice** by Thomas Anderson and Michael Dahlin – Provides a modern, practical perspective on OS design.
3. **Linux Kernel Development** by Robert Love – Excellent for students interested in open-source OS and kernel programming.
4. **Computer Systems: A Programmer's Perspective** by Randal E. Bryant and David R. O'Hallaron – Bridges OS concepts with system-level programming.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15
CO1	3	2	2	2	2	2	2	1	1	2	1	2	1	2	-
CO2	3	2	3	2	3	2	2	2	1	2	2	2	2	3	-
CO3	3	2	3	2	3	2	2	2	2	3	2	3	2	3	-
CO4	3	3	3	3	3	3	3	2	2	3	2	3	3	3	-
CO5	2	3	3	2	3	3	2	2	2	3	2	2	3	3	-
CO6	2	2	2	2	2	3	3	3	3	3	3	3	3	2	-
CO7	3	3	3	3	3	3	3	3	3	3	3	3	2	3	-

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

CO1: Understanding Operating System Modules

CO1: PO1: Understanding OS modules builds strong foundational and integrated computing knowledge.

CO1: PO2: Learning OS structure develops systematic procedural understanding of system functions.

CO1: PO3: Analyzing OS components enhances logical thinking and problem-solving ability.

CO1: PO4: Explaining OS architecture improves technical communication skills.

CO1: PO5: Evaluating system components strengthens analytical judgment.

CO1: PO10: Understanding OS design supports system planning and solution development.

CO1: PO11: Learning responsible system usage promotes ethical awareness in computing.

CO1: PO14: Studying OS modules strengthens specialized subject knowledge in Computer Applications.

CO2: Process Scheduling

CO2: PO1: Learning scheduling algorithms enhances integrated knowledge of resource management.

CO2: PO2: Studying scheduling techniques develops structured procedural skills.

CO2: PO3: Comparing scheduling algorithms strengthens analytical and problem-solving capabilities.

CO2: PO5: Selecting suitable algorithms improves decision-making skills.

CO2: PO10: Designing efficient scheduling strategies supports system-level solution development.

CO2: PO14: Applying scheduling concepts strengthens specialized computing knowledge.

CO3: System Calls and OS Interaction

CO3: PO1: Understanding system calls builds integrated knowledge of system and application interaction.

CO3: PO2: Learning system call procedures enhances structured technical skill development.

CO3: PO3: Analyzing OS-application communication improves logical reasoning.

CO3: PO4: Explaining system-level interactions enhances professional communication skills.

CO3: PO10: Implementing system-level solutions supports system planning competencies.

CO3: PO11: Understanding controlled system access promotes ethical computing practices.

CO3: PO14: Applying system call concepts strengthens specialized subject expertise.

CO4: Memory Management

CO4: PO1: Studying memory management broadens foundational computing knowledge.

CO4: PO2: Learning memory allocation techniques builds procedural competence.

CO4: PO3: Solving memory allocation problems enhances analytical thinking.

CO4: PO5: Evaluating memory strategies strengthens decision-making skills.

CO4: PO10: Designing efficient memory systems supports solution development for real-world computing problems.

CO4: PO14: Applying memory management concepts strengthens domain-specific expertise.

CO5: Process Synchronization

CO5: PO1: Understanding synchronization mechanisms enhances comprehensive system knowledge.

CO5: PO2: Learning synchronization algorithms builds structured technical skills.

CO5: PO3: Resolving concurrency issues strengthens logical problem-solving ability.

CO5: PO5: Evaluating synchronization methods improves analytical judgment.

CO5: PO10: Designing coordination mechanisms supports system development skills.

CO5: PO13: Understanding cooperative processes promotes teamwork and collaborative skills.

CO5: PO14: Applying synchronization concepts strengthens specialized computing knowledge.

CO6: Deadlock Management

CO6: PO1: Studying deadlock concepts deepens integrated system knowledge.

CO6: PO2: Learning deadlock handling techniques develops procedural expertise.

CO6: PO3: Analyzing deadlock conditions enhances critical problem-solving skills.

CO6: PO5: Evaluating prevention and avoidance strategies strengthens decision-making ability.

CO6: PO10: Designing deadlock-free systems supports system planning and development competencies.

CO6: PO13: Understanding resource sharing concepts promotes collaborative system understanding.

CO6: PO14: Applying deadlock management techniques strengthens specialized domain knowledge.

CO7: Advanced Memory Management

CO7: PO1: Exploring advanced memory techniques enhances holistic and integrated computing knowledge.

CO7: PO2: Studying paging and segmentation develops structured procedural skills.

CO7: PO3: Solving advanced memory problems strengthens analytical capability.

CO7: PO5: Comparing memory management strategies improves decision-making skills.

CO7: PO10: Designing optimized memory systems supports complex solution development.

CO7: PO14: Applying advanced memory concepts strengthens specialized subject expertise.

**CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science)
(2024 Pattern)**

Name of the Programme : B.C.A. (Science)

Program Code : BCA
Class : T.Y.B.C.A
Semester : V
Course Type : Major Elective [Theory]
Course Name : Block chain Technology
Course Code : BCA-308-MJE(A)
No. of Credits : 02
No. of Teaching Hours : 30

Prerequisites:

Understanding of Problem solving techniques using a programming language and basic data structures.

Course Objectives:

1. Understand the fundamental concepts, history, and need for blockchain technology.
2. Explain the cryptographic principles that secure blockchain networks.
3. Analyze blockchain architecture, components, and consensus mechanisms.
4. Distinguish between different types of blockchain platforms and cryptocurrencies.
5. Develop basic smart contracts and understand decentralized application (DApp) development.
6. Evaluate real-world applications of blockchain across industries.
7. Examine security, privacy, legal, and regulatory challenges in blockchain adoption.

Course Outcomes:

At the end of the course, students will be able to:

- CO1:** Explain the fundamental concepts and working principles of Blockchain technology.
CO2: Describe cryptographic techniques such as hashing and digital signatures used in Blockchain.
CO3: Compare different consensus mechanisms like PoW and PoS.
CO4: Analyze Bitcoin and Ethereum Blockchain architectures.
CO5: Develop simple smart contracts using Ethereum platform.
CO6: Identify real-world use cases of Blockchain in various sectors.
CO7: Evaluate security, privacy, and regulatory challenges in Blockchain systems.

Topics and Learning Points

Unit	Title and Contents	No. of Lectures
Unit 1	Introduction to Blockchain <ul style="list-style-type: none"> • History and Evolution of Blockchain • Blockchain Architecture and Components • How does a block chain work? • Types of Blockchain (Public, Private, Consortium) • Structure of a Block • Blockchain vs Traditional Databases • Overview of Bitcoin 	06
Unit 2	Cryptocurrency and Bitcoin <ul style="list-style-type: none"> • What Are Different Blockchain Technologies? • Benefits of using Blockchain Technology. • The Origin of Blockchain Completed. • Blockchain came from Bitcoin. 	08
Unit 3	Consensus Algorithms <ul style="list-style-type: none"> • Cryptography and Cryptographic Algorithms • Transaction Execution and Distribution • Components of Blockchain Ecosystem • Transactions and UTXO Model • Mining Process • Proof of Work (PoW) • Proof of Stake (PoS) 	10
Unit 4	Blockchain Applications and Security <ul style="list-style-type: none"> • Blockchain in Banking and Finance • Blockchain in Supply Chain • Blockchain in Healthcare • Blockchain in Government and E-Governance • Security Issues in Blockchain • Privacy Challenges • Legal and Regulatory Framework • Future Trends in Blockchain Technology 	06

Reference /Text Books:

Text Books:

1. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2016.

2. Antonopoulos, A. M., Mastering Bitcoin, O'Reilly Media, 2nd Edition, 2017.
3. Antonopoulos, A. M., & Wood, G., Mastering Ethereum, O'Reilly Media, 2018.

Reference Books:

1. Bashir, I., Mastering Blockchain, Packt Publishing, 3rd Edition, 2020.
2. Drescher, D., Blockchain Basics, Apress, 2017.
3. Swan, M., Blockchain: Blueprint for a New Economy, O'Reilly Media, 2015.
4. Tapscott, D., & Tapscott, A., Blockchain Revolution, Portfolio, 2016.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15
CO1	3	2	2	2	2	2	2	1	1	2	1	2	1	2	1
CO2	3	2	3	2	3	2	2	2	1	2	2	2	2	3	2
CO3	3	2	3	2	3	2	2	2	2	3	2	3	2	3	3
CO4	3	3	3	3	3	3	3	2	2	3	2	3	3	3	2
CO5	2	3	3	2	3	3	2	2	2	3	2	2	3	3	2
CO6	2	2	2	2	2	3	3	3	3	3	3	3	3	2	2
CO7	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

CO1 → PO1 (3 – Strongly Related):

Explaining fundamental concepts of Blockchain builds strong foundational and integrated technical knowledge.

CO2 → PO1 (3 – Strongly Related):

Understanding cryptographic techniques strengthens core knowledge of security principles and applied mathematics.

CO3 → PO1 (3 – Strongly Related):

Comparing consensus mechanisms enhances analytical understanding of distributed systems concepts.

CO4 → PO1 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures integrates theoretical and practical Blockchain knowledge.

CO5 → PO1 (2 – Moderately Related):

Developing smart contracts applies conceptual knowledge but focuses more on practical implementation skills.

CO6 → PO1 (2 – Moderately Related):

Identifying real-world use cases connects Blockchain knowledge with interdisciplinary applications.

CO7 → PO1 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges requires comprehensive and integrated domain

knowledge.

CO1 → PO2 (2 – Moderately Related):

Understanding Blockchain working principles supports procedural learning but focuses more on conceptual clarity than hands-on skill.

CO2 → PO2 (2 – Moderately Related):

Describing cryptographic techniques develops procedural awareness of security implementation methods.

CO3 → PO2 (2 – Moderately Related):

Comparing consensus mechanisms builds analytical procedure understanding of how Blockchain networks operate.

CO4 → PO2 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures enhances procedural knowledge of system structure and transaction processing.

CO5 → PO2 (3 – Strongly Related):

Developing smart contracts directly builds practical coding and platform-based implementation skills.

CO6 → PO2 (2 – Moderately Related):

Identifying real-world applications supports procedural understanding of how Blockchain solutions are implemented in industries.

CO7 → PO2 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges strengthens procedural skills in risk assessment and compliance handling.

CO1 → PO3 (2 – Moderately Related):

Explaining Blockchain fundamentals supports logical understanding of distributed systems but involves limited problem-solving application.

CO2 → PO3 (3 – Strongly Related):

Understanding hashing and digital signatures enhances logical reasoning in solving security and authentication problems.

CO3 → PO3 (3 – Strongly Related):

Comparing PoW and PoS requires analytical thinking and evaluation of performance, security, and efficiency trade-offs.

CO4 → PO3 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures develops system-level problem-solving and logical assessment skills.

CO5 → PO3 (3 – Strongly Related):

Developing smart contracts demands logical programming and structured problem-solving ability.

CO6 → PO3 (2 – Moderately Related):

Identifying use cases applies logical thinking to match Blockchain features with real-world problems.

CO7 → PO3 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges strengthens critical thinking and decision-making skills.

CO1 → PO4 (2 – Moderately Related):

Explaining Blockchain fundamentals requires clear technical communication of complex concepts.

CO2 → PO4 (2 – Moderately Related):

Describing cryptographic techniques enhances the ability to communicate technical security concepts effectively.

CO3 → PO4 (2 – Moderately Related):

Comparing consensus mechanisms develops skills in presenting analytical comparisons professionally.

CO4 → PO4 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures requires structured technical explanation and presentation skills.

CO5 → PO4 (2 – Moderately Related):

Developing smart contracts involves documenting code logic and communicating implementation details.

CO6 → PO4 (3 – Strongly Related):

Identifying real-world use cases demands effective communication of Blockchain solutions to diverse stakeholders.

CO7 → PO4 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges strengthens professional reporting and technical discussion abilities.

CO1 → PO5 (2 – Moderately Related):

Understanding Blockchain fundamentals supports analytical thinking but involves limited decision-making application.

CO2 → PO5 (3 – Strongly Related):

Analyzing cryptographic techniques strengthens judgment in selecting appropriate security mechanisms.

CO3 → PO5 (3 – Strongly Related):

Comparing PoW and PoS develops decision-making skills by evaluating efficiency, scalability, and security trade-offs.

CO4 → PO5 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures enhances analytical judgment in assessing system design choices.

CO5 → PO5 (3 – Strongly Related):

Developing smart contracts requires logical decision-making in coding structure, gas optimization, and security measures.

CO6 → PO5 (2 – Moderately Related):

Identifying real-world use cases involves evaluating suitability of Blockchain solutions for specific industry problems.

CO7 → PO5 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges demands critical judgment and informed decision-making.

CO1 → PO6 (2 – Moderately Related):

Understanding Blockchain fundamentals builds foundational knowledge that supports innovation and employability.

CO2 → PO6 (2 – Moderately Related):

Knowledge of cryptographic techniques enhances employability in security-focused and Blockchain-based roles.

CO3 → PO6 (2 – Moderately Related):

Comparing consensus mechanisms supports creative thinking in selecting efficient models for innovative solutions.

CO4 → PO6 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures strengthens technical expertise valued in industry and startup ecosystems.

CO5 → PO6 (3 – Strongly Related):

Developing smart contracts directly builds practical skills for Blockchain development and entrepreneurial ventures.

CO6 → PO6 (3 – Strongly Related):

Identifying real-world use cases fosters innovation and entrepreneurial thinking in applying Blockchain across sectors.

CO7 → PO6 (3 – Strongly Related):

Evaluating security and regulatory challenges enhances strategic thinking required for sustainable Blockchain businesses.

CO1 → PO7 (2 – Moderately Related):

Understanding Blockchain fundamentals provides interdisciplinary exposure to computer science, finance, and distributed systems.

CO2 → PO7 (2 – Moderately Related):

Cryptographic techniques connect Blockchain with mathematics, cybersecurity, and information technology domains.

CO3 → PO7 (2 – Moderately Related):

Comparing consensus mechanisms links technical design with economic and energy considerations.

CO4 → PO7 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures integrates knowledge of finance, programming, networking, and economics.

CO5 → PO7 (2 – Moderately Related):

Developing smart contracts applies programming skills to legal, financial, and business-oriented applications.

CO6 → PO7 (3 – Strongly Related):

Identifying real-world use cases demonstrates Blockchain's application across healthcare, supply chain, banking, and governance sectors.

CO7 → PO7 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges requires understanding of legal, ethical, and technological domains.

CO1 → PO8 (1 – Partially Related):

Understanding Blockchain fundamentals has limited direct connection to ethical values but provides basic awareness of transparent systems.

CO2 → PO8 (2 – Moderately Related):

Cryptographic techniques promote data integrity and privacy, supporting ethical handling of information.

CO3 → PO8 (2 – Moderately Related):

Comparing consensus mechanisms includes consideration of fairness, transparency, and energy consumption ethics.

CO4 → PO8 (2 – Moderately Related):

Analyzing Blockchain architectures highlights transparency and decentralization, which support ethical governance models.

CO5 → PO8 (2 – Moderately Related):

Developing smart contracts encourages ethical coding practices and responsible automation of agreements.

CO6 → PO8 (3 – Strongly Related):

Identifying real-world use cases emphasizes Blockchain's role in social welfare, financial inclusion, and transparent public services.

CO7 → PO8 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges directly addresses ethical responsibilities in technology deployment.

CO1 → PO9 (1 – Partially Related):

Understanding Blockchain fundamentals has limited direct linkage to indigenous knowledge but supports modern technological awareness.

CO2 → PO9 (1 – Partially Related):

Cryptographic techniques are primarily technical concepts with minimal connection to indigenous practices.

CO3 → PO9 (2 – Moderately Related):

Consensus mechanisms reflect community-based decision-making principles that can relate to traditional governance systems.

CO4 → PO9 (2 – Moderately Related):

Analyzing Blockchain architectures enables understanding of how modern digital systems can support traditional economic models.

CO5 → PO9 (2 – Moderately Related):

Developing smart contracts can be applied to digitize traditional agreements and community transactions.

CO6 → PO9 (3 – Strongly Related):

Identifying real-world use cases encourages applying Blockchain to preserve indigenous knowledge, land records, and artisan supply chains.

CO7 → PO9 (3 – Strongly Related):

Evaluating security and regulatory challenges supports ethical integration of indigenous rights within contemporary digital frameworks.

CO1 → PO10 (2 – Moderately Related):

Understanding Blockchain fundamentals supports system planning but focuses mainly on conceptual knowledge.

CO2 → PO10 (2 – Moderately Related):

Knowledge of cryptographic techniques aids in designing secure Blockchain-based systems.

CO3 → PO10 (3 – Strongly Related):

Comparing consensus mechanisms helps in selecting appropriate models during system design and planning.

CO4 → PO10 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures strengthens ability to plan and structure Blockchain solutions.

CO5 → PO10 (3 – Strongly Related):

Developing smart contracts directly contributes to solution development and implementation skills.

CO6 → PO10 (3 – Strongly Related):

Identifying real-world use cases supports planning effective Blockchain-based solutions for industry problems.

CO7 → PO10 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges ensures robust and compliant system development.

CO1 → PO11 (1 – Partially Related):

Understanding Blockchain fundamentals provides basic awareness of transparency but has limited direct ethical focus.

CO2 → PO11 (2 – Moderately Related):

Cryptographic techniques promote data integrity and privacy, supporting ethical data protection practices.

CO3 → PO11 (2 – Moderately Related):

Comparing consensus mechanisms includes ethical considerations such as fairness and environmental impact.

CO4 → PO11 (2 – Moderately Related):

Analyzing Blockchain architectures highlights decentralization and transparency that contribute to social accountability.

CO5 → PO11 (2 – Moderately Related):

Developing smart contracts encourages responsible coding and ethical automation of agreements.

CO6 → PO11 (3 – Strongly Related):

Identifying real-world use cases emphasizes Blockchain's role in transparency, anti-corruption, and public welfare systems.

CO7 → PO11 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges directly addresses ethical responsibility and compliance in Blockchain deployment.

CO1 → PO12 (2 – Moderately Related):

Understanding Blockchain fundamentals supports research orientation but focuses mainly on conceptual clarity.

CO2 → PO12 (3 – Strongly Related):

Studying cryptographic techniques strengthens analytical and methodological skills required for security research.

CO3 → PO12 (3 – Strongly Related):

Comparing consensus mechanisms develops research ability through evaluation of performance, scalability, and security metrics.

CO4 → PO12 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures enhances investigative and system-level research competence.

CO5 → PO12 (2 – Moderately Related):

Developing smart contracts builds experimental and practical skills but involves limited research methodology.

CO6 → PO12 (3 – Strongly Related):

Identifying real-world use cases promotes exploratory research and interdisciplinary study of Blockchain applications.

CO7 → PO12 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges requires critical analysis and systematic research skills.

CO1 → PO13 (1 – Partially Related):

Explaining Blockchain fundamentals mainly builds individual conceptual understanding with limited direct team interaction.

CO2 → PO13 (2 – Moderately Related):

Understanding cryptographic techniques supports collaborative security design discussions within development teams.

CO3 → PO13 (2 – Moderately Related):

Comparing consensus mechanisms encourages group analysis and technical decision-making in team environments.

CO4 → PO13 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures often involves collaborative system study and technical teamwork.

CO5 → PO13 (3 – Strongly Related):

Developing smart contracts typically requires coordinated coding, testing, and deployment within development teams.

CO6 → PO13 (3 – Strongly Related):

Identifying real-world use cases promotes teamwork in designing Blockchain solutions for industry problems.

CO7 → PO13 (2 – Moderately Related):

Evaluating security and regulatory challenges involves collaborative risk assessment and compliance planning.

CO1 → PO14 (2 – Moderately Related):

Explaining Blockchain fundamentals applies core domain knowledge but focuses more on understanding than specialization.

CO2 → PO14 (3 – Strongly Related):

Describing hashing and digital signatures requires application of specialized cryptographic knowledge.

CO3 → PO14 (3 – Strongly Related):

Comparing PoW and PoS demonstrates application of advanced distributed system concepts.

CO4 → PO14 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures applies in-depth, specialized Blockchain platform

knowledge.

CO5 → PO14 (3 – Strongly Related):

Developing smart contracts directly applies specialized Ethereum programming skills.

CO6 → PO14 (2 – Moderately Related):

Identifying use cases applies domain knowledge to industry contexts but with moderate specialization depth.

CO7 → PO14 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges applies specialized knowledge of Blockchain governance and cybersecurity.

CO1 → PO15 (1 – Partially Related):

Understanding Blockchain fundamentals has limited direct connection to environmental and risk considerations.

CO2 → PO15 (2 – Moderately Related):

Cryptographic techniques contribute to risk management by ensuring data security and integrity.

CO3 → PO15 (3 – Strongly Related):

Comparing PoW and PoS involves evaluating environmental impact (energy consumption) and associated operational risks.

CO4 → PO15 (2 – Moderately Related):

Analyzing Blockchain architectures supports assessment of system risks and sustainability concerns.

CO5 → PO15 (2 – Moderately Related):

Developing smart contracts requires identifying potential security risks and ensuring reliable execution.

CO6 → PO15 (2 – Moderately Related):

Identifying real-world use cases includes considering sustainable and socially responsible Blockchain applications.

CO7 → PO15 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges directly addresses risk management and responsible technology deployment.

CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science) (2024 Pattern)

Name of the Programme	: B.C.A. (Science)
Program Code	: BCA
Class	: T.Y.B.C.A
Semester	: V
Course Type	: Major Elective [Practical]
Course Name	: Lab Course based on BCA-307-MJE(A)
Course Code	: BCA-307-MJE (B)
No. of Credits	: 02
No. of Teaching Hours	: 60

Prerequisites:

- Basic Knowledge of Computer

Course Objectives:

1. To understand Complexity of Operating system as a software. .
2. To understand design issues related to process management and various related algorithms
3. To understand design issues related to memory management and various related algorithms

Course Outcomes:

At the end of the course, students will be able to:

CO1: Working of commands

CO2: Implementation of process scheduling

CO3: Implementation of deadlock avoidance algorithm

CO4: Study the role of the operating system in managing memory.

CO5: Implement memory management algorithms

CO6: Simulation of Algorithms

CO7: Learn network administration and configuration

Topics and Learning Points

Assignment No.	Name of Assignment	No. of Practical Required
1.	Linux and Windows Commands	01

2.	CPU Scheduling	04
3.	Bankers Algorithm	02
4.	Memory Management	04
5.	Case Study	01

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15
CO1	3	2	2	2	2	2	2	1	1	2	1	2	1	2	-
CO2	3	2	3	2	3	2	2	2	1	2	2	2	2	3	-
CO3	3	2	3	2	3	2	2	2	2	3	2	3	2	3	-
CO4	3	3	3	3	3	3	3	2	2	3	2	3	3	3	-
CO5	2	3	3	2	3	3	2	2	2	3	2	2	3	3	-
CO6	2	2	2	2	2	3	3	3	3	3	3	3	3	2	-
CO7	3	3	3	3	3	3	3	3	3	3	3	3	2	3	-

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

CO1: Understanding Operating System Modules

CO1: PO1: Understanding OS modules builds strong foundational and integrated computing knowledge.

CO1: PO2: Learning OS structure develops systematic procedural understanding of system functions.

CO1: PO3: Analyzing OS components enhances logical thinking and problem-solving ability.

CO1: PO4: Explaining OS architecture improves technical communication skills.

CO1: PO5: Evaluating system components strengthens analytical judgment.

CO1: PO10: Understanding OS design supports system planning and solution development.

CO1: PO11: Learning responsible system usage promotes ethical awareness in computing.

CO1: PO14: Studying OS modules strengthens specialized subject knowledge in Computer Applications.

CO2: Process Scheduling

CO2: PO1: Learning scheduling algorithms enhances integrated knowledge of resource management.

CO2: PO2: Studying scheduling techniques develops structured procedural skills.

CO2: PO3: Comparing scheduling algorithms strengthens analytical and problem-solving capabilities.

CO2: PO5: Selecting suitable algorithms improves decision-making skills.

CO2: PO10: Designing efficient scheduling strategies supports system-level solution development.

CO2: PO14: Applying scheduling concepts strengthens specialized computing knowledge.

CO3: System Calls and OS Interaction

CO3: PO1: Understanding system calls builds integrated knowledge of system and application interaction.

CO3: PO2: Learning system call procedures enhances structured technical skill development.

CO3: PO3: Analyzing OS–application communication improves logical reasoning.

CO3: PO4: Explaining system-level interactions enhances professional communication skills.

CO3: PO10: Implementing system-level solutions supports system planning competencies.

CO3: PO11: Understanding controlled system access promotes ethical computing practices.

CO3: PO14: Applying system call concepts strengthens specialized subject expertise.

CO4: Memory Management

CO4: PO1: Studying memory management broadens foundational computing knowledge.

CO4: PO2: Learning memory allocation techniques builds procedural competence.

CO4: PO3: Solving memory allocation problems enhances analytical thinking.

CO4: PO5: Evaluating memory strategies strengthens decision-making skills.

CO4: PO10: Designing efficient memory systems supports solution development for real-world computing problems.

CO4: PO14: Applying memory management concepts strengthens domain-specific expertise.

CO5: Process Synchronization

CO5: PO1: Understanding synchronization mechanisms enhances comprehensive system knowledge.

CO5: PO2: Learning synchronization algorithms builds structured technical skills.

CO5: PO3: Resolving concurrency issues strengthens logical problem-solving ability.

CO5: PO5: Evaluating synchronization methods improves analytical judgment.

CO5: PO10: Designing coordination mechanisms supports system development skills.

CO5: PO13: Understanding cooperative processes promotes teamwork and collaborative skills.

CO5: PO14: Applying synchronization concepts strengthens specialized computing knowledge.

CO6: Deadlock Management

CO6: PO1: Studying deadlock concepts deepens integrated system knowledge.

CO6: PO2: Learning deadlock handling techniques develops procedural expertise.

CO6: PO3: Analyzing deadlock conditions enhances critical problem-solving skills.

CO6: PO5: Evaluating prevention and avoidance strategies strengthens decision-making ability.

CO6: PO10: Designing deadlock-free systems supports system planning and development competencies.

CO6: PO13: Understanding resource sharing concepts promotes collaborative system understanding.

CO6: PO14: Applying deadlock management techniques strengthens specialized domain knowledge.

CO7: Advanced Memory Management

CO7: PO1: Exploring advanced memory techniques enhances holistic and integrated computing knowledge.

CO7: PO2: Studying paging and segmentation develops structured procedural skills.

CO7: PO3: Solving advanced memory problems strengthens analytical capability.

CO7: PO5: Comparing memory management strategies improves decision-making skills.

CO7: PO10: Designing optimized memory systems supports complex solution development.

CO7: PO14: Applying advanced memory concepts strengthens specialized subject expertise.

**CBCS Syllabus as per NEP 2020 for T.Y.B.C.A. (Science)
(2024 Pattern)**

Name of the Programme	: B.C.A. (Science)
Program Code	: BCA
Class	: T.Y.B.C.A
Semester	: V
Course Type	: Major Elective [Practical]
Course Name	: Lab Course On BCA-308-MJE(B)
Course Code	: BCA-308-MJE-(B)
No. of Credits	: 02
No. of Practical Hours	: 60

Prerequisites:

- Understanding of Problem solving techniques using a programming language and basic data structures.

Course Objectives:

1. Understand the fundamental concepts, history, and need for blockchain technology.
2. Explain the cryptographic principles that secure blockchain networks.
3. Analyze blockchain architecture, components, and consensus mechanisms.
4. Distinguish between different types of blockchain platforms and cryptocurrencies.
5. Develop basic smart contracts and understand decentralized application (DApp) development.
6. Evaluate real-world applications of blockchain across industries.
7. Examine security, privacy, legal, and regulatory challenges in blockchain adoption.

Course Outcomes:

At the end of the course, students will be able to:

- CO1:** Explain the fundamental concepts and working principles of Blockchain technology.
- CO2:** Describe cryptographic techniques such as hashing and digital signatures used in Blockchain.
- CO3:** Compare different consensus mechanisms like Proof of Work (PoW) and Proof of Stake (PoS).
- CO4:** Analyze Bitcoin and Ethereum Blockchain architectures.
- CO5:** Develop simple smart contracts using the Ethereum platform.
- CO6:** Identify real-world use cases of Blockchain in various sectors.
- CO7:** Evaluate security, privacy, and regulatory challenges in Blockchain systems.

Topics and Learning Points

Sr. No.	Assignment Name	No. of Assignment
Assignment 1	Introduction to Blockchain (Block Structure Exercise, Database Comparison, Blockchain Types Case Study)	03
Assignment 2	Cryptocurrency and Bitcoin (Bitcoin Wallet Demo, Transaction Flow Mapping, Cryptocurrency Comparison)	03
Assignment 3	Consensus Algorithms (Hashing Practice, Proof of Work Simulation, Consensus Debate)	03
Assignment 4	Blockchain Applications and Security (Supply Chain Use Case, Smart Contract Development, Security & Privacy Analysis)	03

Text Books:

1. **Bitcoin and Cryptocurrency Technologies** Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder Princeton University Press, 2016
2. **Mastering Bitcoin (2nd Edition)** Andreas M. Antonopoulos O'Reilly Media, 2017
3. **Mastering Ethereum** Andreas M. Antonopoulos, Gavin Wood O'Reilly Media, 2018

Reference Books:

1. **Mastering Blockchain (3rd Edition)** Imran Bashir Packt Publishing, 2020
2. **Blockchain Basics: A Non-Technical Introduction in 25 Steps** Daniel Drescher Apress, 2017
3. **Blockchain: Blueprint for a New Economy** Melanie Swan O'Reilly Media, 2015

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15
CO1	3	2	2	2	2	2	2	1	1	2	1	2	1	2	1
CO2	3	2	3	2	3	2	2	2	1	2	2	2	2	3	2
CO3	3	2	3	2	3	2	2	2	2	3	2	3	2	3	3
CO4	3	3	3	3	3	3	3	2	2	3	2	3	3	3	2

CO5	2	3	3	2	3	3	2	2	2	3	2	2	3	3	2
CO6	2	2	2	2	2	3	3	3	3	3	3	3	3	2	2
CO7	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

CO1 → PO1 (3 – Strongly Related):

Explaining fundamental concepts of Blockchain builds strong foundational and integrated technical knowledge.

CO2 → PO1 (3 – Strongly Related):

Understanding cryptographic techniques strengthens core knowledge of security principles and applied mathematics.

CO3 → PO1 (3 – Strongly Related):

Comparing consensus mechanisms enhances analytical understanding of distributed systems concepts.

CO4 → PO1 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures integrates theoretical and practical Blockchain knowledge.

CO5 → PO1 (2 – Moderately Related):

Developing smart contracts applies conceptual knowledge but focuses more on practical implementation skills.

CO6 → PO1 (2 – Moderately Related):

Identifying real-world use cases connects Blockchain knowledge with interdisciplinary applications.

CO7 → PO1 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges requires comprehensive and integrated domain knowledge.

CO1 → PO2 (2 – Moderately Related):

Understanding Blockchain working principles supports procedural learning but focuses more on conceptual clarity than hands-on skill.

CO2 → PO2 (2 – Moderately Related):

Describing cryptographic techniques develops procedural awareness of security implementation methods.

CO3 → PO2 (2 – Moderately Related):

Comparing consensus mechanisms builds analytical procedure understanding of how Blockchain networks operate.

CO4 → PO2 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures enhances procedural knowledge of system structure and transaction processing.

CO5 → PO2 (3 – Strongly Related):

Developing smart contracts directly builds practical coding and platform-based implementation skills.

CO6 → PO2 (2 – Moderately Related):

Identifying real-world applications supports procedural understanding of how Blockchain solutions are implemented in industries.

CO7 → PO2 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges strengthens procedural skills in risk assessment and compliance handling.

CO1 → PO3 (2 – Moderately Related):

Explaining Blockchain fundamentals supports logical understanding of distributed systems but involves limited problem-solving application.

CO2 → PO3 (3 – Strongly Related):

Understanding hashing and digital signatures enhances logical reasoning in solving security and authentication problems.

CO3 → PO3 (3 – Strongly Related):

Comparing PoW and PoS requires analytical thinking and evaluation of performance, security, and efficiency trade-offs.

CO4 → PO3 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures develops system-level problem-solving and logical assessment skills.

CO5 → PO3 (3 – Strongly Related):

Developing smart contracts demands logical programming and structured problem-solving ability.

CO6 → PO3 (2 – Moderately Related):

Identifying use cases applies logical thinking to match Blockchain features with real-world problems.

CO7 → PO3 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges strengthens critical thinking and decision-making skills.

CO1 → PO4 (2 – Moderately Related):

Explaining Blockchain fundamentals requires clear technical communication of complex concepts.

CO2 → PO4 (2 – Moderately Related):

Describing cryptographic techniques enhances the ability to communicate technical security concepts effectively.

CO3 → PO4 (2 – Moderately Related):

Comparing consensus mechanisms develops skills in presenting analytical comparisons professionally.

CO4 → PO4 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures requires structured technical explanation and presentation skills.

CO5 → PO4 (2 – Moderately Related):

Developing smart contracts involves documenting code logic and communicating implementation details.

CO6 → PO4 (3 – Strongly Related):

Identifying real-world use cases demands effective communication of Blockchain solutions to diverse stakeholders.

CO7 → PO4 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges strengthens professional reporting and technical discussion abilities.

CO1 → PO5 (2 – Moderately Related):

Understanding Blockchain fundamentals supports analytical thinking but involves limited decision-making application.

CO2 → PO5 (3 – Strongly Related):

Analyzing cryptographic techniques strengthens judgment in selecting appropriate security mechanisms.

CO3 → PO5 (3 – Strongly Related):

Comparing PoW and PoS develops decision-making skills by evaluating efficiency, scalability, and security trade-offs.

CO4 → PO5 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures enhances analytical judgment in assessing system design choices.

CO5 → PO5 (3 – Strongly Related):

Developing smart contracts requires logical decision-making in coding structure, gas optimization, and security measures.

CO6 → PO5 (2 – Moderately Related):

Identifying real-world use cases involves evaluating suitability of Blockchain solutions for specific industry problems.

CO7 → PO5 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges demands critical judgment and informed decision-

making.

CO1 → PO6 (2 – Moderately Related):

Understanding Blockchain fundamentals builds foundational knowledge that supports innovation and employability.

CO2 → PO6 (2 – Moderately Related):

Knowledge of cryptographic techniques enhances employability in security-focused and Blockchain-based roles.

CO3 → PO6 (2 – Moderately Related):

Comparing consensus mechanisms supports creative thinking in selecting efficient models for innovative solutions.

CO4 → PO6 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures strengthens technical expertise valued in industry and startup ecosystems.

CO5 → PO6 (3 – Strongly Related):

Developing smart contracts directly builds practical skills for Blockchain development and entrepreneurial ventures.

CO6 → PO6 (3 – Strongly Related):

Identifying real-world use cases fosters innovation and entrepreneurial thinking in applying Blockchain across sectors.

CO7 → PO6 (3 – Strongly Related):

Evaluating security and regulatory challenges enhances strategic thinking required for sustainable Blockchain businesses.

CO1 → PO7 (2 – Moderately Related):

Understanding Blockchain fundamentals provides interdisciplinary exposure to computer science, finance, and distributed systems.

CO2 → PO7 (2 – Moderately Related):

Cryptographic techniques connect Blockchain with mathematics, cybersecurity, and information technology domains.

CO3 → PO7 (2 – Moderately Related):

Comparing consensus mechanisms links technical design with economic and energy considerations.

CO4 → PO7 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures integrates knowledge of finance, programming, networking, and economics.

CO5 → PO7 (2 – Moderately Related):

Developing smart contracts applies programming skills to legal, financial, and business-oriented applications.

CO6 → PO7 (3 – Strongly Related):

Identifying real-world use cases demonstrates Blockchain's application across healthcare, supply chain, banking, and governance sectors.

CO7 → PO7 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges requires understanding of legal, ethical, and technological domains.

CO1 → PO8 (1 – Partially Related):

Understanding Blockchain fundamentals has limited direct connection to ethical values but provides basic awareness of transparent systems.

CO2 → PO8 (2 – Moderately Related):

Cryptographic techniques promote data integrity and privacy, supporting ethical handling of information.

CO3 → PO8 (2 – Moderately Related):

Comparing consensus mechanisms includes consideration of fairness, transparency, and energy

consumption ethics.

CO4 → PO8 (2 – Moderately Related):

Analyzing Blockchain architectures highlights transparency and decentralization, which support ethical governance models.

CO5 → PO8 (2 – Moderately Related):

Developing smart contracts encourages ethical coding practices and responsible automation of agreements.

CO6 → PO8 (3 – Strongly Related):

Identifying real-world use cases emphasizes Blockchain's role in social welfare, financial inclusion, and transparent public services.

CO7 → PO8 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges directly addresses ethical responsibilities in technology deployment.

CO1 → PO9 (1 – Partially Related):

Understanding Blockchain fundamentals has limited direct linkage to indigenous knowledge but supports modern technological awareness.

CO2 → PO9 (1 – Partially Related):

Cryptographic techniques are primarily technical concepts with minimal connection to indigenous practices.

CO3 → PO9 (2 – Moderately Related):

Consensus mechanisms reflect community-based decision-making principles that can relate to traditional governance systems.

CO4 → PO9 (2 – Moderately Related):

Analyzing Blockchain architectures enables understanding of how modern digital systems can support traditional economic models.

CO5 → PO9 (2 – Moderately Related):

Developing smart contracts can be applied to digitize traditional agreements and community transactions.

CO6 → PO9 (3 – Strongly Related):

Identifying real-world use cases encourages applying Blockchain to preserve indigenous knowledge, land records, and artisan supply chains.

CO7 → PO9 (3 – Strongly Related):

Evaluating security and regulatory challenges supports ethical integration of indigenous rights within contemporary digital frameworks.

CO1 → PO10 (2 – Moderately Related):

Understanding Blockchain fundamentals supports system planning but focuses mainly on conceptual knowledge.

CO2 → PO10 (2 – Moderately Related):

Knowledge of cryptographic techniques aids in designing secure Blockchain-based systems.

CO3 → PO10 (3 – Strongly Related):

Comparing consensus mechanisms helps in selecting appropriate models during system design and planning.

CO4 → PO10 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures strengthens ability to plan and structure Blockchain solutions.

CO5 → PO10 (3 – Strongly Related):

Developing smart contracts directly contributes to solution development and implementation skills.

CO6 → PO10 (3 – Strongly Related):

Identifying real-world use cases supports planning effective Blockchain-based solutions for industry problems.

CO7 → PO10 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges ensures robust and compliant system development.

CO1 → PO11 (1 – Partially Related):

Understanding Blockchain fundamentals provides basic awareness of transparency but has limited direct ethical focus.

CO2 → PO11 (2 – Moderately Related):

Cryptographic techniques promote data integrity and privacy, supporting ethical data protection practices.

CO3 → PO11 (2 – Moderately Related):

Comparing consensus mechanisms includes ethical considerations such as fairness and environmental impact.

CO4 → PO11 (2 – Moderately Related):

Analyzing Blockchain architectures highlights decentralization and transparency that contribute to social accountability.

CO5 → PO11 (2 – Moderately Related):

Developing smart contracts encourages responsible coding and ethical automation of agreements.

CO6 → PO11 (3 – Strongly Related):

Identifying real-world use cases emphasizes Blockchain's role in transparency, anti-corruption, and public welfare systems.

CO7 → PO11 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges directly addresses ethical responsibility and compliance in Blockchain deployment.

CO1 → PO12 (2 – Moderately Related):

Understanding Blockchain fundamentals supports research orientation but focuses mainly on conceptual clarity.

CO2 → PO12 (3 – Strongly Related):

Studying cryptographic techniques strengthens analytical and methodological skills required for security research.

CO3 → PO12 (3 – Strongly Related):

Comparing consensus mechanisms develops research ability through evaluation of performance, scalability, and security metrics.

CO4 → PO12 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures enhances investigative and system-level research competence.

CO5 → PO12 (2 – Moderately Related):

Developing smart contracts builds experimental and practical skills but involves limited research methodology.

CO6 → PO12 (3 – Strongly Related):

Identifying real-world use cases promotes exploratory research and interdisciplinary study of Blockchain applications.

CO7 → PO12 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges requires critical analysis and systematic research skills.

CO1 → PO13 (1 – Partially Related):

Explaining Blockchain fundamentals mainly builds individual conceptual understanding with limited direct team interaction.

CO2 → PO13 (2 – Moderately Related):

Understanding cryptographic techniques supports collaborative security design discussions within development teams.

CO3 → PO13 (2 – Moderately Related):

Comparing consensus mechanisms encourages group analysis and technical decision-making in team environments.

CO4 → PO13 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures often involves collaborative system study and technical teamwork.

CO5 → PO13 (3 – Strongly Related):

Developing smart contracts typically requires coordinated coding, testing, and deployment within development teams.

CO6 → PO13 (3 – Strongly Related):

Identifying real-world use cases promotes teamwork in designing Blockchain solutions for industry problems.

CO7 → PO13 (2 – Moderately Related):

Evaluating security and regulatory challenges involves collaborative risk assessment and compliance planning.

CO1 → PO14 (2 – Moderately Related):

Explaining Blockchain fundamentals applies core domain knowledge but focuses more on understanding than specialization.

CO2 → PO14 (3 – Strongly Related):

Describing hashing and digital signatures requires application of specialized cryptographic knowledge.

CO3 → PO14 (3 – Strongly Related):

Comparing PoW and PoS demonstrates application of advanced distributed system concepts.

CO4 → PO14 (3 – Strongly Related):

Analyzing Bitcoin and Ethereum architectures applies in-depth, specialized Blockchain platform knowledge.

CO5 → PO14 (3 – Strongly Related):

Developing smart contracts directly applies specialized Ethereum programming skills.

CO6 → PO14 (2 – Moderately Related):

Identifying use cases applies domain knowledge to industry contexts but with moderate specialization depth.

CO7 → PO14 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges applies specialized knowledge of Blockchain governance and cybersecurity.

CO1 → PO15 (1 – Partially Related):

Understanding Blockchain fundamentals has limited direct connection to environmental and risk considerations.

CO2 → PO15 (2 – Moderately Related):

Cryptographic techniques contribute to risk management by ensuring data security and integrity.

CO3 → PO15 (3 – Strongly Related):

Comparing PoW and PoS involves evaluating environmental impact (energy consumption) and associated operational risks.

CO4 → PO15 (2 – Moderately Related):

Analyzing Blockchain architectures supports assessment of system risks and sustainability concerns.

CO5 → PO15 (2 – Moderately Related):

Developing smart contracts requires identifying potential security risks and ensuring reliable execution.

CO6 → PO15 (2 – Moderately Related):

Identifying real-world use cases includes considering sustainable and socially responsible Blockchain applications.

CO7 → PO15 (3 – Strongly Related):

Evaluating security, privacy, and regulatory challenges directly addresses risk management and responsible technology deployment.

**CBCS Syllabus as per NEP 2020 for T.Y.BCA (Computer Application)
(2024Pattern)**

Name of the Programme	: BCA (Computer Application)
Programme Code	: BCA
Class	: T.Y.BCA
Semester	: V
Course Type	: Minor [Theory]
Course Code	: BCA-310-MN
Course Title	: E-Commerce concepts and business model
No. of Credits	: 02
No. of Teaching Hours	: 30

Course Objectives:

1. Understand WordPress as a CMS.
2. Design and develop websites using themes and plugins.
3. Manage content and maintain website security.
4. Apply WordPress for real-world applications like blogs and business sites.
5. Understand E-commerce on business models and strategy

Course Outcomes:

At the end of the course, students will be able to:

CO1: Explain the features, structure, and functionality of WordPress as a CMS.

CO2: Install, configure, and manage and handle WordPress websites on both localhost and live hosting environments.

CO3: Design responsive websites using pre-built themes and custom templates also Integrate plugins and widgets to enhance website functionality and appearance.

CO4: Manage and publish website content effectively using the WordPress dashboard.

CO5: Apply SEO techniques, security measures, and backup strategies for optimal website performance.

CO6: Develop, deploy, and maintain complete WordPress-based projects suitable for academic or professional use

CO7: Analyze the impact of E-commerce on business models and strategy and Capable to evaluate the effectiveness of e-commerce strategies

Topics and Learning Points

Units	Title and Contents	No. of Lectures
1	Introduction 1.1 Overview of E-Commerce 1.2 Applications of E-Commerce 1.3 Types of E-Commerce 1.4 Elements of E-Commerce 1.5 Electronic Data Interchange (EDI) 1.6 Electronic Payment Systems	07
2	Introduction To WordPress 1.1 What is WordPress? Overview and History 1.2 Understanding CMS and WordPress 1.3 WordPress.org vs. WordPress.com 1.4 Installing WordPress (Localhost & Manual Installation) 1.5 Overview of WordPress Dashboard & General Settings 1.6 WordPress Themes and Plugins 1.7 WordPress Pages vs Posts. 1.8 Creating and Managing Menus, Categories, and Tags	07
3	Blog and Searching Fundamentals 1.1 Concept of Blog 1.2 Characteristics of Blog 1.3 Blog Development 1.4 Search Engine 1.5 Multilingual Search 1.6 Translation of search results	06
4	Web Hosting and E-Commerce and Mobile Commerce(M-Commerce) 1.1 Introduction to Web Hosting 1.2 Hosting Web Site on Web Server 1.3 E-Commerce Models 1.4 E-Commerce Frame Work 1.5 Elements of E-Commerce Infrastructure 1.6 Introduction to M-Commerce 1.7 Types of M-Commerce 1.8 Services of M-Commerce 1.9 Benefits and Limitations of M-Commerce 1.10 Mobile Business Services 1.11 Wireless Application Protocol	10

Reference /Text Books:

Text Books

1. **Fundamentals of E-Commerce** by Dr. Subhabrata De – A revised edition aligned with CBCS & NEP curriculum, widely used in Indian universities for B.Com, BBA, and BCA courses.
2. **E-Commerce: Business, Technology, Society** by Kenneth C. Laudon & Carol Guercio Traver – A globally recognized textbook covering both technical and managerial aspects of e-commerce.
3. **Electronic Commerce: A Managerial Perspective** by Efraim Turban, David King, Jae Lee, Ting-Peng Liang, and Deborrah Turban – Focuses on strategies, technologies, and applications of e-commerce.

Reference Books:

1. **E-Commerce Masters Manual: Ultimate Guide to Selling Anything Online** by Amar Bedi – Practical guide for building, scaling, and automating e-commerce brands.
2. **E-Commerce Brand Building: Build a Billion Dollar E-Commerce Brand from Scratch** by Rishi Kandari – Focused on entrepreneurial and brand-building aspects of online business.
3. **E-Commerce Essentials** by Ralph M. Stair & George W. Reynolds – Provides foundational knowledge for students new to e-commerce.
4. **Digital Business and E-Commerce Management** by Dave Chaffey – Covers digital strategies, online marketing, and e-commerce management practices.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15
CO1	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1
CO2	2	3	2	1	2	2	1	1	1	2	1	1	1	2	1
CO3	2	3	3	2	2	3	2	1	1	3	1	2	2	3	1
CO4	1	3	2	3	2	2	2	1	1	2	1	1	2	2	1
CO5	2	2	3	2	3	2	2	2	1	3	3	2	1	2	2
CO6	2	3	3	2	3	3	2	2	2	3	2	2	3	3	2
CO7	3	2	3	2	3	3	3	2	2	2	2	3	2	3	2

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO with All CO'S

PO1: COs collectively integrate E-Commerce concepts, WordPress development, and business strategy knowledge.

PO2: CO2, CO3, CO4, and CO6 strongly emphasize procedural and hands-on skill development.

PO3: CO3, CO5, CO6, and CO7 develop logical thinking and problem-solving through real-world platform design and analysis.

PO4: CO4 and CO6 enhance professional communication via content management, documentation, and presentations.

PO5: CO5, CO6, and CO7 promote analytical judgment in SEO, security decisions, and E-Commerce strategy evaluation.

PO6: CO3, CO6, and CO7 strongly support employability and entrepreneurship through platform creation and business insight.

PO7: CO7 links computing knowledge with management, marketing, and interdisciplinary business understanding.

PO8: CO5 and CO6 partially address ethical and social responsibility through secure, responsible digital practices.

PO9: CO6 and CO7 moderately integrate traditional business knowledge with modern E-Commerce technologies.

PO10: CO3, CO5, and CO6 strongly support system planning, development, and solution implementation.

PO11: CO5 and CO6 emphasize ethical awareness related to data security, privacy, and professional accountability.

PO12: CO7 and CO6 support research skills through analysis of E-Commerce models and project-based learning.

PO13: CO6 strongly supports teamwork through collaborative WordPress project development.

PO14: CO3, CO6, and CO7 apply specialized knowledge in computer applications and E-Commerce domains.

PO15: CO5 and CO6 moderately address environmental responsibility through optimized, sustainable digital systems.