Bachelor of Computer Applications

Semester–IV

Course Code	Course Title	L	T	P	Credits
BCA-122A	Programming with Python (Pre-requisite: None)	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-122A.1	Understand the fundamentals of Python programming, including data types, input/output
BCA-122A.2	Analyze and implement operators and control structure in Python.
BCA-122A.3	Utilize ordered data objects including string, list and tuple.
BCA-122A.4	Utilize unordered data objects including set and dictionary and design functions.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-122A.1	3	-	-	2	-	-	-	2
BCA-122A.2	3	3	2	2	-	-	-	3
BCA-122A.3	3	3	3	3	-	-	-	3
BCA-122A.4	3	3	3	3	-	-	2	3

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 9**

Basics of Python

Overview and applications areas of Python, Python installation, Comments, Keywords, Identifiers, Variables declaration, Standard Data types, id(), type(), quotations for string literals, data type conversion methods, Input/Output statements, escape characters, round( ), importing 'math' package

UNIT-II **Contact Hours: 11**

Operators and Control statements in Python

Operators: Arithmetic, Assignment, Comparison, Logical, Bitwise, Reference or identity (is/ is not), Membership (in /not in)

Control Statements: conditional statements (if, else, elif), loop statements (for, while), break, continue, pass statement.

UNIT-III **Contact Hours: 13**

Ordered Data Objects

Ordered Data Objects: String: declaration, indexing, len(), concatenation, repetition, 'in'/'not in' operator,

slice, Traversal, Built-in functions; List: declaration, indexing, len(), creating list using range(), list with initial values, converting string to list, concatenation, repetition, 'in'/'not in' operator, slice, traversal, built-in functions; Tuple: declaration, indexing, len( ), creating tuple using range(), tuple with initial values, converting string to tuple, converting list to tuple, concatenation, repetition, 'in'/'not in' operator, slice, traversal, built-in functions.

UNIT-IV **Contact Hours: 12**

Unordered Data Objects and Functions

Unordered Data Objects: Set: Roster and Rule form, len(), converting list to set and set to list, Built-in functions, Set operations; Dictionary: declaration, len(), inserting and accessing a value, keys(), values(), items(), Built-in functions

Functions: defining a function, calling a function, keyword arguments, default arguments, variable length arguments, scope of variables, Recursive function, map(), Lambda function

Text Books:

1. E. Balagurusamy, *Introduction to Computing and Problem Solving Using Python*, McGrawHill Education
2. Yashavant Kanetkar, Aditya Kanetkar, *Let us Python*, BPB Publications
3. Bob Dowling, *An introduction to Python for absolute beginners*, Cambridge University Press.
4. Vamsi Kurama, *Python Programming: A Modern Approach*, Pearson Education
5. Sheetal Taneja, Naveen Kumar, *Python Programming A Modular Approach*, Pearson
6. Rao R. Nageswara , *Core Python Programming*, Dream Tech, New Delhi

Reference Books:

1. Mark Lutz, *Learning Python*, 5th edition, O'Reilly.
2. Martin C. Brown, *Python: The Complete Reference*, McGraw-Hill

Course Code	Course Title	L	T	P	Credits
BCA-272A	**Programming with Python Lab** (Pre-requisite: Python)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-272A.1	Develop Python programs using data types, input/output
BCA-272A.2	Apply operators and control structures in Python
BCA-272A.3	Utilize ordered data objects in designing Python programs
BCA-272A.4	Design programs using unordered data objects and functions.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-272A.1	3	2	3	2	-	-	-	2
BCA-272A.2	3	3	3	2	-	-	-	3
BCA-272A.3	3	3	3	2	-	-	-	3
BCA-272A.4	3	3	3	2	-	-	2	3

List of Experiments

No.	Experiment Detail
1.	Assume you travel 80 km to and fro in a day. Fuel Cost is INR 80 per liter and your vehicle's Fuel Average is 18 km/litre. Write a Python program to calculate the Driving Cost per day.
2.	Write a program to calculate the area and circumference of a circle for the given radius as input.
3.	Write a program to make use of various Data conversion methods in Python.
4.	Write a Python program which iterates the integers from 1 to 30 (included). if number is divisible by 3 print "PIET", if number is divisible by 5 print "DCA". if number is divisible by both 3 and 5 print "PIET & DCA" and otherwise print the number itself.
5.	Write a Python program to display the following pattern for the given input of no. of lines. 1 121 12321 1234321
6.	Write a program to check whether the given integer number is Palindrome or not.
7.	Assuming your weight in kilogram and height in meters, calculate your Body Mass Index (BMI) value using the formula BMI=weight/(height*height) Height is given in (feet.inches) form and weight is given in kgs and to convert total height in meters consider 1 Feet = 0.3048 meter and 1 Inch = 0.0254 meter
8.	Create a list of city names from the given list where city name contains substring 'pur' cities=['Jaipur','Ambala','Nagpur','Kanpur','Karnal','Sholapur']
9.	Wrtie a program to remove leading and trailing blank spaces from string items in the list animals = [' monkey ', ' pitbull dog ', ' rabbit ']
10.	Book order is given as list of sublists with the following items Order Number Book Title Author Quantity Unit Price

	34587 Learning Python Mark Lutz 4 40.95 98762 Programming Python Mark Lutz 5 56.80 77226 Head First Python Paul Barry 3 32.95 Write a Python program to generate the Order summary in this List of tuple form [('34587', 163.8), ('98762', 284.0), ('77226', 108.85)] Where first item in the tuple is Order Number, second item is Net Price i.e. Quantity*Unit Price if Net Price < 100.00 it should be increased by Rs. 10.
11.	employees=['Gurpreet','Manjot','Shabadpreet','Gurleen','Vishal','Noor','Navjeet'] gym_members=['Shabadpreet','Vishal','Navjeet'] developers=['Shabadpreet','Gurpreet','Gurleen','Manjot','Navjeet'] From the above given lists write Python program to answer the following queries: i) List all employees who have gym membership and also developers ii) List all the employees who are not either gym members or developers
12.	A string is given to you e.g. "aabaacdaadd". Create a dictionary of each character in the given string and its frequency, i.e. output={'a': 6, 'b': 1, 'c': 1, 'd': 3}
13.	In a shop store, data operator enters Item Name, Quantity sold and its Unit Price one by one till 'CLOSE' is entered. Create Python dictionary of Item Name and its Net Price. **Sample Input**: BANANA FRIES 2 20.60 POTATO CHIPS 3 15.50 APPLE JUICE 10 100.75 CANDY 5 2.50 APPLE JUICE 6 100.75 CANDY 5 2.50 CANDY 5 2.50 BANANA FRIES 3 20.60 CANDY 5 2.50 POTATO CHIPS 5 15.50 CLOSE **Sample Output:** {'BANANA FRIES': 103.0, 'POTATO CHIPS': 124.0, 'APPLE JUICE': 1612.0, 'CAND Y': 50.0}
14.	Write a function to make use of variable length arguments.
15.	Write a Recursive function to generate ith term of a Fibonacci Series 0,1,1,2,3,5,8,13,... Generate n terms of Fibonacci Series using this recursive function.

Text Books:

1. E. Balagurusamy, *Introduction to Computing and Problem Solving Using Python*, McGrawHill Education
2. Yashavant Kanetkar, Aditya Kanetkar, *Let us Python*, BPB Publications
3. Bob Dowling, *An introduction to Python for absolute beginners*, Cambridge University Press.
4. Vamsi Kurama, *Python Programming: A Modern Approach*, Pearson Education

5. Sheetal Taneja, Naveen Kumar, *Python Programming A Modular Approach*, Pearson
6. Rao R. Nageswara , *Core Python Programming*, Dream Tech, New Delhi

Reference Books:

1. Mark Lutz, *Learning Python*, 5th edition, O'Reilly.
2. Martin C. Brown, *Python: The Complete Reference*, McGraw-Hill

Course Code	Course Title	L	T	P	Credits
BCA-215A	Full Stack Development-1 (MERN) (Pre-requisite: Web Technology (HTML, CSS, JavaScript))	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-215A.1	Understand the core concepts of the MERN stack, including React.js, Express.js, Node.js, and MongoDB.
BCA-215A.2	Develop front-end applications using React.js, including state management and component-based architecture.
BCA-215A.3	Build RESTful APIs with Express.js and integrate MongoDB for backend data storage and retrieval.
BCA-215A.4	Implement authentication, middleware, and deployment techniques for full-stack applications.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-215A.1	3	-	-	-	-	-	-	2
BCA-215A.2	3	-	3	3	-	2	-	2
BCA-215A.3	3	2	3	3	-	-	-	2
BCA-215A.4	3	2	3	3	-	2	2	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I Contact Hours: 12

Introduction to MERN Stack & Node.js

Overview of Full Stack Development and MERN Stack.Node.js: Introduction, history, features, and its role in MERN. Comparison between JavaScript Client-side vs. Server-side programming. Installation of Node.js and npm, writing and executing basic scripts. Understanding JavaScript fundamentals: Variables, functions, loops, and arrays. Event-driven architecture and the Node.js runtime environment.

UNIT-II Contact Hours: 11

Frontend Development with React.js basics

React.js Introduction: Why React? Virtual DOM, React Setup (Create React App). JSX & Components: Functional vs. Class Components, Props, State. React Hooks: useState, useEffect, useContext, handling side effects. Event Handling & Forms: Controlled vs. Uncontrolled components. React Router: Navigation, dynamic routing, route parameters. State Management in React: Context API, Prop Drilling, Lifting State Up.

UNIT-III Contact Hours: 12

Backend Development with Node.js & Express.js

Node.js Modules: Core modules, global modules, and user-defined modules.Express.js: Introduction, routing,

request handling, middleware. Template Engines (EJS) for rendering dynamic web pages. Middleware: Introduction to Express.js middleware and error handling. Database Integration: Introduction to MongoDB, performing CRUD operations with Mongoose. Using Postman for API testing.

UNIT-IV **Contact Hours: 10**

Advanced Concepts and Deployment

Asynchronous Programming: Callback, Promises, Async/Await, Event loop in Node.js. Mongoose ORM: Schema and model creation, advanced queries. Authentication & Authorization: JSON Web Tokens (JWT), bcrypt, Role-based authentication. State Management in React: Context API, Redux basics. File Handling & APIs: Using `multer` for file uploads, building Search APIs. Deployment: Hosting MERN applications using services like Vercel, Netlify, and AWS.

Text Books:

1. Basarat Ali Syed, *Beginning Node.js*, Apress.
2. Adam Boduch, Roy Derks, *React and React Native*, Packt Publishing.
3. Ethan Brown, *Learning JavaScript Design Patterns*, O'Reilly.
4. Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node,* Apress, 2019.

Reference Books:

1. Boronczyk, Naramore, *Beginning PHP, Apache, MySQL Web Development*, Wiley India Pvt.Ltd.
2. Kyle Simpson, *You Don't Know JS: Up and Going*, O'Reilly.
3. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.
4. Simon Holmes, Clive Harber, *Getting MEAN with Mongo, Express, Angular, and Node*, Manning Publications.

Course Code	Course Title	L	T	P	Credits
BCA-274A	Full Stack Development-1 (MERN) Lab (Pre-requisite: MERN Stack)	0	0	4	2
		CIE	SEE		Total
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-274A.1	Implement Node.js applications
BCA-274A.2	Write programs implementing Express.js
BCA-274A.3	Write programs implementing React.js
BCA-274A.4	Deploy a Full stack development application

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-274A.1	3	2	2	2	-	-	-	2
BCA-274A.2	3	2	2	2	-	-	-	2
BCA-274A.3	3	3	2	2	-	-	-	2
BCA-274A.4	3	3	3	2	-	2	2	2

List of Experiments

No.	Experiment Detail
1.	Write a simple 'Hello World' program in Node.js.
2.	Create a Node.js script that reads and writes files using the `fs` module.
3.	Build a basic Express.js server with different routes.
4.	Implement middleware in Express.js for logging request details.
5.	Create a RESTful API in Express.js to perform CRUD operations on a MongoDB database.
6.	Implement authentication using JWT in an Express.js application.
7.	Upload and retrieve files/images using `multer` in Express.js.
8.	Set up a basic React application and create a functional component.
9.	Implement React state and props in a simple To-Do List app.
10.	Build a multi-page React app using React Router.
11.	Fetch data from an API and display it using React (Axios or Fetch API).
12.	Implement global state management using the Context API in React.
13.	Implement form validation and handle user input in React.
14.	Connect a React frontend with a Node.js/Express backend using Axios.
15.	Deploy a full-stack MERN application (frontend + backend) using Vercel and Netlify.

Text Books:

1. Basarat Ali Syed, *Beginning Node.js*, Apress.
2. Adam Boduch, Roy Derks, *React and React Native*, Packt Publishing.
3. Ethan Brown, *Learning JavaScript Design Patterns*, O'Reilly.
4. Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, Apress, 2019.

Reference Books:

1. Boronczyk, Naramore, *Beginning PHP, Apache, MySQL Web Development*, Wiley India Pvt.Ltd.
2. Kyle Simpson, *You Don't Know JS: Up and Going*, O'Reilly.
3. David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly Media.
4. Simon Holmes, Clive Harber, *Getting MEAN with Mongo, Express, Angular, and Node*, Manning Publications.

Course Code	Course Title	L	T	P	Credits
BCA-202A	Quantitative Aptitude (Pre-requisite: Basic math)	2	0	0	2
		CIE	SEE		Total
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-202A.1	Understand the concept of system of algebraic equations and clocks.
BCA-202A.2	Solve problems related to time-distance, work-time, and work-wages.
BCA-202A.3	Solve problems related to interest, partnership, sets, and trigonometric ratios.
BCA-202A.4	Apply the concepts of Permutations and Combinations, Probability, and Data interpretation

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-202A.1	3	-	-	-	-	-	-	2
BCA-202A.2	3	2	3	-	-	-	-	3
BCA-202A.3	3	2	3	-	2	-	-	3
BCA-202A.4	3	2	3	-	-	-	-	3

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 8**

System of algebraic equations, Ages and Clocks problems

Linear Equations, Quadratic equations, System of algebraic equations in two variables and their applications in simple problems. Problems on ages, Clocks.

UNIT-II **Contact Hours: 7**

Time and distance, Work and Time problems

Time and distance: Problems based on trains, Boats and Streams, Pipes and Cistern. Work and time: Problems on work and time, work and wages.

UNIT-III **Contact Hours: 7**

Interest, Partnership, Sets, Height and Distance problems

Simple interest, Compound Interest, Partnership. Basic idea of set theory to solve practical problems. Trigonometric ratios and identities, Height and distance.

UNIT-IV **Contact Hours: 8**

Permutations and Combinations, Data interpretation

Basic idea of Permutations and Combinations. Events and sample space, Probability. Data interpretation: Raw and grouped data, Bar Graph, Pie Chart, Mean, Median and Mode.

Text Books:

1. R. S. Aggarwal, *Quantitative Aptitude*, S. Chand & Company Limited, New Delhi
2. A. Guha. *Quantitative Aptitude*, 7th Edition, McGraw-Hill Publications
3. V. Dyke, J. Rogers and H. Adams, *Fundamentals of Mathematics*, Cengage Learning

Reference Books:

1. A.S. Tussy, R. D. Gustafson and D. Koenig, *Basic Mathematics for College Students*, Brooks Cole
2. C. C. Pinter, *A Book of Set Theory*, Dover Publications

Course Code	Course Title	L	T	P	Credits
BCA-204A	**Animation** (Pre-requisite: None)	2	0	0	2
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-204A.1	Understand the fundamentals and historical development of animation.
BCA-204A.2	Apply techniques of 2D animation including sketching, still life, and digital layout creation.
BCA-204A.3	Use Adobe Photoshop for image editing, painting, and storyboarding in animation.
BCA-204A.4	Understand to Create simple 2D animations

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-204A.1	3	-	-	-	-	-	-	-
BCA-204A.2	3	-	2	2	-	-	-	2
BCA-204A.3	3	-	2	3	-	-	-	2
BCA-204A.4	3	2	2	2	-	2	-	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 7**

Introduction, History and Basics of Animation

Definition and scope of animation, History and evolution of animation, Traditional vs. computer animation, Principles of animation: squash and stretch, timing, anticipation, staging, follow-through, Applications in media, advertising, education, gaming

UNIT-II **Contact Hours: 7**

2D Animation and Drawing Techniques

Introduction to 2D animation: Basics of sketching, still life, assignment of basic drawing, Composition of basic elements, creating digital layout, working with visual images, Animate (Scanning, tracing, ink and paint)

UNIT-III **Contact Hours: 8**

Adobe Photoshop for Animation and Design

Drawing concepts in digital media, working in different media: drawing, collage, painting, Photoshop interface and tools, working with layers, frames, timeline, Professional image editing & manipulation, Storyboarding and visual storytelling, understanding background composition, Exporting GIFs and videos.

UNIT-IV **Contact Hours: 8**

Animation tools

Animation: Interface, drawing tools and symbols, Motion and shape tweens, working with layers and scenes, Simple project: making a short-animated clip

Text Books:

1. Williams, Richard, *The Animator's Survival Kit*, Faber & Faber.
2. White, Tony, *Animation from Pencils to Pixels: Classical Techniques for the Digital Animator*, Focal Press.
3. Adobe Creative Team, *Adobe Photoshop Classroom in a Book*, Adobe Press.

Reference Books:

1. Blair, Preston, *Cartoon Animation*, Walter Foster Publishing.
2. Glebas, Francis, *Directing the Story: Professional Storytelling and Storyboarding Techniques for Live Action and Animation*, Focal Press.
3. Patmore, Chris, *The Complete Animation Course: The Principles, Practice and Techniques of Successful Animation*, Barron's Educational Series.

Course Code	Course Title	L	T	P	Credits
BCA-276A	**Animation Lab** (Pre-requisite: Animation Tools)	0	0	4	2
		CIE	**SEE**		**Total**
		50	50		100

Course Outcomes (COs): At the end of this course, students will be able to

BCA-276A.1	Understand the history, growth, and fundamental principles of animation.
BCA-276A.2	Develop hands-on skills in traditional and digital 2D animation techniques.
BCA-276A.3	Learn and apply Photoshop tools for visual design, editing, and storyboarding.
BCA-276A.4	Create engaging short animated clips using animation software.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-276A.1	3	-	-	-	-	-	-	-
BCA-276A.2	3	-	2	2	-	-	-	3
BCA-276A.3	3	-	3	3	-	2	-	3
BCA-276A.4	3	2	3	3	-	2	-	3

List of Experiments

No.	Experiment Detail
1.	Explore the journey of animation by researching its history and evolution. Present your findings in a creative report.
2.	Understand and demonstrate key animation principles like squash and stretch, anticipation, staging, and timing using simple illustrations.
3.	Analyze real-world uses of animation across media, advertising, education, and gaming with examples. Create a short presentation.
4.	Practice basic sketching by drawing simple still life objects. Focus on shapes, proportions, and shading.
5.	Learn to compose visual elements and create a digital layout. Use simple forms to build a scene.
6.	Scan hand-drawn sketches and apply digital tracing, inking, and coloring techniques to bring them to life.
7.	Get familiar with the Photoshop interface. Use basic tools to draw and color a simple illustration.
8.	Create a digital composition combining drawing, painting, and collage tools in Photoshop. Let your creativity flow!
9.	Plan and design a storyboard for a short-animated sequence. Use layers and timeline in Photoshop to visualize the story.
10.	Create a simple animated background scene and learn how to export it in GIF and video formats.
11.	Explore the interface of animation software and understand how to use essential drawing tools and symbols.
12.	Create motion and shape tweens to animate a bouncing ball or moving object. Practice smooth transitions.

13.	Work with multiple layers and scenes to develop a basic animation sequence. Understand scene transitions.
14.	Design and animate a simple character using symbols. Experiment with motion paths and basic effects.
15.	Bring everything together by developing a short-animated clip as your final mini project. Tell a story using all you've learned!

Text Books:

1. Williams, Richard, *The Animator's Survival Kit*, Faber & Faber.
2. White, Tony, *Animation from Pencils to Pixels: Classical Techniques for the Digital Animator*, Focal Press.
3. Adobe Creative Team, *Adobe Photoshop Classroom in a Book*, Adobe Press.

Reference Books:

1. Blair, Preston, *Cartoon Animation*, Walter Foster Publishing.
2. Glebas, Francis, *Directing the Story: Professional Storytelling and Storyboarding Techniques for Live Action and Animation*, Focal Press.
3. Patmore, Chris, *The Complete Animation Course: The Principles, Practice and Techniques of Successful Animation*, Barron's Educational Series.

Course Code	Course Title	L	T	P	Credits
BCA-252A(i)	High Performance Networks System (Pre-requisite: None)	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-252A(i).1	Understand the fundamental concepts and requirements of high-performance networks.
BCA-252A(i).2	Analyze different switching techniques and their performance characteristics.
BCA-252A(i).3	Explain and compare various network performance metrics and congestion control mechanisms.
BCA-252A(i).4	Explore advanced technologies such as MPLS, QoS, and software-defined networking (SDN).

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-252A(i).1	3	-	-	-	-	-	-	2
BCA-252A(i).2	3	3	2	2	-	-	-	3
BCA-252A(i).3	3	2	2	-	-	-	-	3
BCA-252A(i).4	3	3	3	2	-	2	2	3

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT-I **Contact Hours: 12**

Introduction to High Performance Networks

Network evolution and performance trends, Requirements of high-speed networks, Performance metrics: throughput, latency, delay, jitter, Technologies overview: Broadband ISDN, SONET/SDH, Gigabit Ethernet, Packet vs Circuit switching

UNIT-II **Contact Hours: 11**

Switching and Transmission Technologies

Switching techniques: Store-and-forward, Cut-through, Fast packet switching and ATM networks, Optical networking and Dense Wavelength Division Multiplexing (DWDM), Wireless high-speed technologies: LTE, 5G, Network interface technologies and adapters

UNIT-III **Contact Hours: 12**

Congestion Control and QoS Mechanisms

Congestion causes and effects, Congestion control algorithms: Leaky bucket, Token bucket, Flow control: TCP/IP and ATM-based mechanisms,Quality of Service (QoS) requirements, QoS models: IntServ, DiffServ,Traffic shaping and engineering.

UNIT-IV **Contact Hours: 10**

Advanced Topics and Future Trends

MPLS (Multi-Protocol Label Switching), SDN (Software Defined Networking) and NFV (Network Function Virtualization),Network virtualization and slicing, High-performance data centers and cloud networking, Emerging trends: Edge computing, IoT networking, and quantum networks

Text Books:

1. William Stallings, *High-Speed Networks and Internets: Performance and Quality of Service*, Pearson
2. Behrouz A. Forouzan, *Data Communications and Networking,* McGraw-Hill
3. James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, Pearson

Reference Books:

1. Uyless Black, *High-Performance Networks: Technology and Protocols*, Prentice Hall
2. Jean Walrand and Pravin Varaiya, *High-Performance Communication Networks,* Morgan Kaufmann
3. Larry L. Peterson and Bruce S. Davie, *Computer Networks: A Systems Approach,* Morgan Kaufmann

Course Code	Course Title	L	T	P	Credits
BCA-252A(ii)	**Compiler Design** (Pre-requisite: None)	3	0	0	3
		CIE	**SEE**		**Total**
		40	60		100

Course Outcomes (COs): At the end of this course, students will be able to	
BCA-252A(ii).1	Design various finite automata with and without output and interpret Regular languages
BCA-252A(ii).2	Design, analyze and interpret Context Free languages, Expression and Grammars
BCA-252A(ii).3	Understand the basic functioning of Compiler and its tools
BCA-252A(ii).4	To demonstrate the use of intermediate code generation.

Course Outcomes (CO) to Program Outcomes (PO) mapping (scale 1: Low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
BCA-252A(ii).1	3	-	2	-	-	-	-	2
BCA-252A(ii).2	3	3	2	-	-	-	-	2
BCA-252A(ii).3	3	-	-	-	-	-	-	2
BCA-252A(ii).4	3	2	2	2	-	-	-	2

Instructions for Paper Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

UNIT – I **Contact Hours: 12**

Finite State Machines and Regular Expressions

Finite State Machines: Finite Automata, Designing of DFA and NDFA, NFA with E-Transitions, Equivalence of DFA and NFA with proof, Minimization of FA

Regular Expressions and Regular languages, Laws of Regular Expressions, Kleene's Theorem 1 and 2, Properties and Limitations of FSM.

UNIT – II **Contact Hours: 11**

FSM with Output and Formal Grammars

FSM with Output: Moore and Mealy Machines, Arden's Theorem with proof, Closure Properties of Regular Sets, Pumping Lemma for Regular Grammars.

Formal Grammars: Definition, Construction of Regular & Context Free Grammar, Derivation, Parse Trees, Ambiguity, Removal of Ambiguity, Simplification of Context Free Grammar, CNF and GNF, Closure properties of CFL.

UNIT – III **Contact Hours: 10**

Translator and Compiler

Introduction To Translator and Compiler, Difference between Interpreter, Assembler and Compiler, Overview and Use of Linker and Loader, Types of Compilers, Phases of Compiler, Bootstrapping, Compiler Construction Tools.

UNIT – IV **Contact Hours: 12**

Lexical Analysis and Intermediate code generation

Lexical Analysis: Role of Lexical Analyzer, Design of Lexical Analyzer, Specification and Recognition of Tokens, Language for Specifying Lexical Analyzers, Lex tool, Input Buffering, Implementation of Lexical Analyzer.

Intermediate code generation: Intermediate forms of source programs– abstract syntax tree, polish notation and three address code, types of three address statements and its implementation

Text Books:

1. Hopcroft, Ullman, "*Introduction to Automata Theory, Languages and Computation*", Pearson Education.
2. Alfred V Aho, *Principles of Compiler Design*, Narosa Publishing House.
3. Jean Paul Tremblay and Sorenson, *The Theory and Practice of Compiler Writing*, McGraw Hill.
4. Alfread V. AHO, Ravi Sethi & J.D. Ullman, *Compilers Principle, Techniques & Tools*, Addison Wesley

Reference Books:

1. K.L.P. Mishra and N.Chandrasekaran, "*Theory of Computer Science Automata, Languages and Computation*", PHI.
2. Tremblay & Sorenson, *Theory and practice of compiler writing*, Mc. Graw Hill.
3. M. Joseph, *Elements Compiler Design*, University Science Press
4. Fischer, *Crafting a Compiler in C*, Pearson Education.