Karmaveer Bhaurao Patil University, Satara

Syllabus for

B. Sc. II (Computer Science)

Under

Faculty of Science and Technology

(As per NEP 2020)

With effect from Academic Year 2025-2026

Syllabus for Bachelor of Science (Computer Science) Part II

PREAMBLE:

Bachelor of Science is an integrated academic degree in the faculty of science. The revision of existing syllabus of Computer Science subject in Science Faculty is essential. This is a humble endeavor to initiate the process towards an era of knowledge. The students from science faculty should also be competent for this change in the technology. In this year, a student will able to handle computers, develop the programs in languages and other peripherals with confidence. In the subject, the student will also get a basic and proper knowledge in the field of Programming skills.

GENERAL OBJEVTIVES OF THE COURSE:

1. To learn basics of Computer, hardware, software, networking.
2. To inculcate the software development attitude and generate interest in the field of Technology.
3. To develop programming skills, management skills, writing skills, Project Analysis skill among students.
4. To inculcate research attitude among students.

PROGRAMME OUTCOMES:

1. Demonstrate a solid understanding of core principles, theories, and concepts in computer science.
2. Design, implement, test, and debug software using widely adopted programming languages.
3. Develop dynamic, interactive web applications using modern web technologies and frameworks.
4. Communicate effectively in both written and verbal forms within multidisciplinary teams.
5. Collaborate efficiently to solve complex problems through teamwork and shared expertise

PROGRAM SPECIFIC OUTCOMES (PSO's):

1. Ability to design, develop, implement computer programs and use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.
2. To work with and communicate effectively with professionals in various fields and

pursue lifelong professional development in computing.

3. To analyse, design, develop, test and apply management principles, mathematical foundations in the development of computational solutions, make them to expert in designing the computer software and hardware.

4. Develop their skills to solve problems in the broad area of programming concepts and appraise environmental and social issues with ethics and manage different projects in inter- disciplinary field.

1. TITLE**:** COMPUTER SCIENCE

2. YEAR OF IMPLEMENTATION: **2025-2026**

3. DURATION: **01 year**

4. PATTERN: **Semester examination**

5. MEDIUM OF INSTRUCTION: **English**

6. COURSE STRUCTURE

Sem Level	**Courses**			**OE**	**VSC/ SEC**	**AEC/ VEC/ IKS**	**OJT/ FP/ CEP/ CC/RP**	**Total Credits**	**Degree/ Cum. Cr. MEME**
	Course I	**Course II**	**Course III**						
	Major	Minor	-						
Sem III (5.0)	Major V (2) Major VI (2) Major P III (2)	Minor V (2) Minor VI (2) Minor P III (2)	-	OE III (2)	VSC I (2) (P) (Major Specific) SEC I (2) (T+P)	AEC I (2) (P) (English) IKS II (2) (Major Specific)	-	22	UG Diploma: 88
Sem IV (5.0)	Major VII (2) Major VIII (2) Major P IV (2)	Minor VII (2) Minor VIII (2) Minor P IV (2)	-	OE IV (2)	VSC II (2) (P) (Major Specific) SEC II (2) (T+P)	AEC II (2) (P) (English) VEC II (2) (Environmental Studies)	-	22	Exit option: 4 Cr. NSQF/ Internship/ Skill Courses
Credits	12	12	-	04	08	08	-	44	

7. COURSE TITLES

B. Sc. (Computer Science) Part-II

Semester: III					
Sr. No.	**Components**		**Course Code**	**Name of the Paper**	**Credits**
1	Major	Paper I	BCST 231	Data Structure using C	2
		Paper II	BCST 232	Data Communication and Computer Network	2
		Practical	BCSP 233	Lab Based on BCST 231 and BCST 232	2
2	Minor	Paper I	BCST 234	Computer Organizations	2
		Paper II	BCST 235	Basics of HTML	2
		Practical	BCSP 236	Lab Based on BCST 234 and BCST 235	2
3	Open Elective		BCSTOE 3	Open Elective III	2
4	VSC-I		BCSTVSC-I	Object Oriented Programming using C++	2
5	SEC-I		BCSTSEC-I	System Analysis and Design	2
6	AEC-I		BCSTAEC-I	English	2
7	IKS-II		BCSTIKS-II	History of Computers in India	2
Total					**22**

Semester: IV					
Sr. No.	**Components**		**Course Code**	**Name of the Paper**	**Credits**
1	Major	Paper I	BCST 241	Object Oriented Concept using JAVA	2
		Paper II	BCST 242	Operating Systems	2
		Practical	BCSP 243	Lab Based on BCST 241 and BCST 242	2
2	Minor	Paper I	BCST 244	Cyber Security	2
		Paper II	BCST 245	Design Principles using CSS	2
		Practical	BCSP 246	Lab Based on BCST 244 and BCST 245	2
3	Open Elective		BCSTOE 4	Open Elective IV	2
4	VSC-II		BCSTVSC-II	Adobe Photoshop	2
5	SEC-II		BCSTSEC-II	XML Programming	2
6	AEC-II		BCSTAEC-II	English	2
7	VEC-II		BCSTVEC-II	Environmental Studies	2
Total					**22**

8. Evaluation Structure:

Semester III (5.0)

Course	Course Category	Course Code	Internal Evaluation			ESE	Total Marks	Credits
			CCE-I	Mid - Semester	CCE-II			
Major	T	BCST 231	05	10	05	30	50	02
	T	BCST 232	05	10	05	30	50	02
	P	BCSP 233	--	--	--	50	50	02
Minor	T	BCST 234	05	10	05	30	50	02
	T	BCST 235	05	10	05	30	50	02
	P	BCSP 236	--	--	--	50	50	02
OE	T	BCSTOE 3 (For IDS Courses)	05	--	05	15	25	01
	P	BCSPOE 3	--	--	--	25	25	01
OE	T	BCSTOE 3 (For Humanities)	05	10	05	30	50	02
VSC	P	BCSPVSC 1	--	--	--	50	50	02
SEC	T	BCSTSEC 1 Theory	05	--	05	15	25	01
	P	BCSPSEC 1 Practical	--	--	--	25	25	01
AEC I	T	BCSTAEC 1	05	10	05	30	50	02
IKS II	T	BCSTIKS 2	05	10	05	30	50	02
Total							**550**	**22**

Semester IV

Course	Course Category	Course Code	Internal Evaluation			ESE	Total Marks	Credits
			CCE-I	Mid - Semester	CCE-II			
Major	T	BCST 241	05	10	05	30	50	02
	T	BCST 242	05	10	05	30	50	02
	P	BCSP 243	--	--	--	50	50	02
Minor	T	BCST 244	05	10	05	30	50	02
	T	BCST 245	05	10	05	30	50	02
	P	BCSP 246	--	--	--	50	50	02
OE	T	BCSTOE 4 (For IDS Courses)	05	--	05	15	25	01
	P	BCSPOE 4	--	--	--	25	25	01
OE	T	BCSTOE 4 (For Humanities)	05	10	05	30	50	02
VSC	P	BCSPVSC 2	--	--	--	50	50	02
SEC	T	BCSTSEC 2 Theory	05	--	05	15	25	01
	P	BCSPSEC 2 Practical	--	--	--	25	25	01
AEC II	T	BCSTAEC 2	05	10	05	30	50	02
VEC II	T	BCSTVEC 2	05	10	05	30	50	02
Total							550	22

OTHER FEATURES:

A) LIBRARY:

Reference books, Textbooks, journals, and Periodicals are available in Institute and Departmental Library. (Separate reference lists are attached along with the respective course syllabus)

B) EQUIPMENT:

a) Computers, LCD projector, smart board

B. Sc. Part II, Semester III Computer Science

BCST 231: Data Structure using C

(Total Credit-02)

Course objectives: By the end of the Course Students will be able to

1. Understand the basic concepts such as Linear and Non-Linear Data structures.
2. Apply the notations used to analyze the Performance of algorithms.
3. Imbibe the behavior of data structures such as stacks queues and their representations.
4. Study an appropriate data structure for a specified application and to understand and analyze various algorithms

UNIT	Content	No. of hours per unit
I	**Basic of Data Structures and Algorithms C**	08
	1.1 Data Structures Basics: Structure and Problem-Solving 1.2 Algorithm Specification-Introduction, Performance analysis- time complexity and space complexity. 1.3 Asymptotic Notation-Big O, Omega and Theta notations, Complexity Analysis Examples, 1.4 Introduction to Linear and Non-Linear data structures, Applications of linear data structure (Searching & Sorting).	
II	**Stack**	07
	2.1 Introduction of stack 2.2 Representation-static & dynamic, Operations. 2.3 Application - infix to postfix & prefix, postfix evaluation 2.4 Concept of Multiple stacks.	
III	**Queue**	07
	3.1 Introduction of Queue. 3.2 Representation -static &dynamic, Operations.	

	3.3 Circular queue, De-Queue, priority queues 3.4 Concept of Multiple Queues	
IV	**Linked List** 4.1 Introduction, Representation of linked lists in Memory, Memory allocation and Garbage collection 4.2 Types of linked list 4.3 Operations on singly linked list, traversing a linked list, Searching a linked list 4.4 Insertion into linked list, Deletion from a linked list.	**08**

Course outcomes: By the end of the Course Students will be able to**:**

1. Prepare and perform an installation of windows server 2008 and identify the various types of file systems and components.
2. Learn fundamentals of C and ability to choose appropriate data structure to represent data items in real world problems
3. Analyze the time and space complexities of algorithms
4. Demonstrate practical understanding of data communication concepts such as error detection, flow control, and transmission techniques.

References

1. Goodrich, Michael T., Roberto Tamassia, and David M. Mount. *Data Structures and Algorithms in C*. Wiley, 2018.
2. Balagurusamy, E. *Data Structures Using C*. McGraw Hill Education, 2017.
3. Perry, Greg. *C Programming Absolute Beginner's Guide*. Que Publishing, 2016.
4. Hanly, Jeri R., and Elliot B. Koffman. *Problem Solving and Program Design in C*. Pearson, 2015.
5. Salaria, R. S. *Modern Data Structures Using C*. Khanna Publishing, 2017.
6. Loudon, Kyle. *Mastering Algorithms with C*. O'Reilly Media, 2018.
7. Karumanchi, Narasimha. *Data Structures and Algorithmic Thinking with Python*. CareerMonk Publications, 2015.

B. Sc. Part II Semester III Computer Science

BCST 232: Introduction to Data Communication and Networks

(Total Credit-02)

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

1. To introduce the fundamental concepts of data communication, including transmission media.
2. To understand the architecture, components, and operations of computer networks.
3. To explore the layered approach to network design (OSI and TCP/IP models).
4. To examine network security, congestion control, and wireless networking principles.

Unit	Content	No. of hours per unit
I	**Introduction to Data Communication and Networks** 1.1 Basics of Data Communication 1.2 Components and Types of Networks (LAN, MAN, WAN) 1.3 Network Topologies 1.4 OSI and TCP/IP Reference Models 1.5 Networking Devices (Hub, Switch, Router, etc.) 1.6 Transmission Modes (Simplex, Half Duplex, Full Duplex)	**08**
II	**Physical and Data Link Layers** 2.1 Transmission Media: Wired & Wireless 2.2 Analog vs Digital Transmission 2.3 Multiplexing: FDM, TDM, WDM 2.4 Error Detection and Correction (Parity, CRC, Checksum) 2.5 Flow Control and Error Control Techniques 2.6 Data Link Layer Protocols: HDLC, PPP 2.7 MAC Protocols: CSMA/CD, CSMA/CA	**07**
III	**Network and Transport Layers** 3.1 Logical Addressing: IPv4 & IPv6 3.2 Subnetting and Super netting 3.3 Routing Concepts and Algorithms (Distance Vector, LinkState) 3.4 Protocols: IP, ICMP, ARP, RARP 3.5 TCP vs UDP 3.6 Connection Management and Congestion Control	**07**

IV	**Application Layer and Network Security** 4.1 Application Layer Protocols: DNS, FTP, HTTP, SMTP, POP3,IMAP 4.2 Client-Server and Peer-to-Peer Models 4.3 Basics of Network Security 4.4 Encryption, Firewalls, VPNs 4.5 Introduction to Cloud and Wireless networks.	**08**

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

1. Understand the basic concepts of data communication and networking, including types of networks
2. Explain the functions and services of each layer in the OSI and TCP/IP reference models.
3. Analyze and evaluate various data encoding, error detection, and correction techniques.
4. Demonstrate knowledge of switching techniques, routing algorithms.

References

1. Stallings, William. *Data and Computer Communications*. Boston: Pearson, 2013.
2. Forouzan, Behrouz A. *Data Communications and Networking*. New York: McGraw-Hill Education, 2012.
3. Kurose, James F., and Keith W. Ross. *Computer Networking: A Top-Down Approach*. Boston: Pearson, 2017.
4. Tanenbaum, Andrew S., and David J. Wetherall. *Computer Networks*. Boston: Pearson, 2013.
5. Comer, Douglas E. *Computer Networks and Internets*. Pearson, 2018.
6. Peterson, Larry L., and Bruce S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann, 2011.
7. Halsall, Fred. *Computer Networking and the Internet*. Pearson Education, 2006.

B. Sc. Part – II, Semester III Computer Science

BCSP 233: Practical's based on BCST231 Data Structure using C and BCST232 Data Communication and Computer Network

(Total Credits-02)

Course objectives: By the end of the course students should be able to…

1. Understand the basic concepts such as Linear and Non-Linear Data structures.
2. Apply the notations used to analyze the Performance of algorithms.
3. Imbibe the behavior of data structures such as stacks queues and their representations.
4. Study an appropriate data structure for a specified application and to understand and analyze various algorithms.

Practical No	Content	No. of hours per practical
1	Write a linear searching algorithm with example.	04
2	Write a binary searching algorithm with example.	04
3	Write a Bubble sort, Insertion sort , selection sort algorithm with example.	04
4	Problems with Big O notations omega notations, Theta notations	04
5	STACK Implementation using Array with PUSH, POP, and display Operations.	04
6	STACK Implementation using Link list with PUSH, POP, and display Operations	04
7	Perform Queues operations using simple Array implementation. Use Templates.	04
8	Perform Queues operations using link list implementation. Use Templates.	04
9	Create and perform different operations on Double-ended Queues using array implementation.	04
10	Create and perform different operations on Double-ended Queues using Linked List implementation.	04
11	Study of Networking Devices – Identify and understand routers, switches, hubs, bridges, and modems.	04
12	IP Address Configuration – Assign static and dynamic IPs to computers.	04
13	Networking Commands – Use commands like ping, ipconfig, tracert, nslookup, netstat, etc.	04
14	Network Topology Setup (Simulation) – Create star, bus, and ring topologies using tools like Cisco Packet Tracer or GNS3.	04

15	Calculate and create subnets from given IPs.	04
16	Static & Dynamic Routing – Setup static routing in Cisco Packet Tracer; observe dynamic routing with RIP or OSPF	04
17	Stop-and-Wait Protocol Simulation – Simple implementation to show flow control.	04
18	File Transfer using FTP – Setup an FTP server and connect clients to transfer files.	04
19	Network Security Tools – Introduction to tools like Nmap, Wireshark, or Cain & Abel for ethical learning.	04
20	Web Server Configuration – Setup Apache or Nginx server to host a website.	04

Course outcomes: By the end of the Course Students will be able to…

1. Prepare and perform an installation of windows server 2008 and identify the various types of file systems and components.
2. Learn fundamentals of C and ability to choose appropriate data structure to represent data items in real world problems
3. Analyze the time and space complexities of algorithms
4. Demonstrate practical understanding of data communication concepts such as error detection, flow control, and transmission techniques.

References

1. Stallings, William. “Data and Computer Communications”. Boston: Pearson, 2013
2. Kurose, James F., and Keith W. Ross. “Computer Networking: A Top-Down Approach”. Boston: Pearson, 2017
3. Tanenbaum, Andrew S., and David J. Wetherall. “Computer Networks”. Boston: Pearson, 2013
4. Goodrich, Michael T., Roberto Tamassia, and David M. Mount. *“Data Structures and Algorithms in C”.* 2018
5. Balagurusamy, E. *“Data Structures Using C”.* 2017
6. Perry, Greg. *“C Programming Absolute Beginner's Guide”.* 2016
7. Hanly, Jeri R. *“Problem Solving and Program Design in C”.* 2015

B. Sc. Part II, Semester III Computer Science III

BCST 234: Computer Organizations

Course (Total Credit-02)

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

1. Define basic concepts and terminology of computers.

2. To understand operate desktop computers to carry out computational tasks.

3. Learn working of Hardware and Software and the importance of operating systems.

4. To identify programming languages, number systems, peripheral devices, networking, multimedia and internet concepts.

Unit	Content	No. of hours per unit
I	**Introduction to Computer Architecture** 1.1Basic Computer Components: CPU, Memory, I/O devices,Basic functions of each component. 1.2 Simple Block Diagram: How data flows within a computer system, Relationship between CPU, Memory, and I/O devices 1.3 Computer Generations: A brief overview of the history and evolution of computers	08
II	**Memory Organization** 2.1 Memory Basics: Types of memory: RAM, ROM, Cache Memory hierarchy: From Registers to Hard Disk 2.2 Cache Memory: What is cache memory and why is it used? Simple explanation of cache hit/miss 2.3 Simple Memory Structure: How memory is organized in a computer system, Addressing in memory (Direct and Indirect addressing)	07
III	**CPU Organization** 3.1 CPU Components: Basic parts of the CPU: ALU (Arithmetic and Logic Unit), Control Unit, Registers, The role of the ALU in processing data 3.2 Instruction Cycle: The Fetch-Decode-Execute cycle, A simple overview of how instructions are processed step by step 3.3 Registers: What are registers? Purpose of general-purpose and	07

	special-purpose registers	
IV	**Input/output Organization** 4.1 Basic I/O Devices: Common input devices (keyboard, mouse) and output devices (monitor, printer) 4.2 I/O Operations: Basic I/O operations: Input, Output, and the role of the CPU 4.3 I/O Interface: How the CPU interacts with I/O devices via ports and buses	**08**

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

1. Describe basic concepts and terminology of information technology.

2. Evaluate the fundamentals of personal computers and their operations.

3. Maintain their small account using the computers and enjoy in the world of Information Technology.

4. Use the computer for basic purposes of preparing his personnel/business letters, viewing information on internet (the web), sending mails, preparing his business presentations, playing games etc.

References

1.Guy Hart-Davis(2023) "The ABCs of Microsoft Office 97 Professional edition", BPB Publications.
2. Karl Schwartz(1998), "Microsoft Windows 98 Training Guide" BPB Publications.
3. C.S. French(1998) "Data Processing and Information Technology", BPB Publications
4. P.K Sinha (1992)`Computer Fundamentals`, BPB Publications

B. Sc. Part II Semester III Computer Science

BCST 235: Basics of HTML

(Total Credit-02)

Course objectives: By the end of the Course Students will be able to

1. Understand HTML structure and syntax.
2. Create responsive and accessible web designs.
3. Insert text, images, links, lists, and tables.
4. Build simple forms and apply attributes.

UNIT	Content	No. of hours per unit
I	**Introduction to HTML** 1.1 Definition and purpose of HTML, Evolution of HTML versions (HTML4, XHTML, HTML 5 etc.). 1.2.Setting up a development environment (text editor, browser) 1.3. HTML Documents, Basic structure of an HTML document. 1.4 Common HTML Tags(Headings, Paragraphs, Line breaks, Horizontal rules). 1.4 HTML Tags, formatting tags, Creating an HTML document. 1.5 Applications of HTML 1.6 Features and Limitations Of HTML.	**08**
II	**Exploring HTML Elements and Functions** 2.1 Golden rules of web designing, Introduction to elements of HTML 2.2 Working with Text, HTML Comments 2.3 Working with Lists, Tables and Frames. 2.4 Formatting tables using colspan, rowspan, and CSS 2.5 Working with Hyperlinks, Images and Multimedia 2.6 Working with Forms and controls, HTML Styles	**07**

	2.7 Linking to external resources (images, CSS files, other web pages)	
III	**Introduction to Cascading Style Sheet** 3.1 Introduction of CSS 3.2 Creating Style Sheet 3.3 CSS Properties 3.4 CSS Styling (Background, Text Format, Controlling Fonts) 3.5 CSS Colors 3.6 Working with block elements and objects 3.7 Working with Lists and Tables.	**07**
IV	**CSS Box Model Fundamentals** 4.1 Box Model (Introduction, Border properties, Padding Properties, Margin properties) 4.2 CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector) 4.3Creating page Layout and Site Designs	**08**

Course outcomes: By the end of the Course Students will be able to**:**

1. To Design and develop basic web pages using HTML..
2. Students will learn how to format text, create hyperlinks, insert images, and embed multimedia content in HTML.
3. Develop responsive designs that adapt to various screen sizes using CSS media queries and flexible units.
4. Students will develop the skills to optimize HTML code for accessibility, search engine optimization (SEO), and cross-browser compatibility.

References

1. Jon Duckett (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
2. Jennifer Niederst Robbins (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics* (5th Edition). O'Reilly Media.
3. Jennifer Niederst Robbins (2013). *HTML5 Pocket Reference* (5th Edition). O'Reilly

Media.
4. Elisabeth Robson & Eric Freeman (2012). *Head First HTML and CSS*. O'Reilly Media.
5. David Sawyer McFarland (2015). *CSS: The Missing Manual* (4th Edition). O'Reilly Media.
6. Ben Henick (2010). *HTML & CSS: The Good Parts*. O'Reilly Media.

B. Sc. Part – II, Semester III Computer Science

BCSP 236: Practical's based on BCST234 Computer Organizations

and

BCST235 Basics of HTML

(Total Credits-02)

Course objectives: By the end of the Course students should be able to..

1. Understand the structure and syntax of HTML to build basic web pages.
2. Apply HTML tags to format text, add images, links, lists, tables, and forms.
3. Develop simple static web pages using proper coding practices.
4. Describe the basic structure and functional units of a computer system.

Practical No	Content	No.of hours per practical
1	Draw the computer's block diagram.	04
2	Draw the computer's block diagram.	04
3	Draw the computer's block diagram.	04
4	Draw the computer's block diagram.	04
5	Draw the computer's block diagram.	04
6	Draw the computer's block diagram.	04
7	Draw the computer's block diagram.	04
8	Simulate basic ALU operations (addition and subtraction).	04
9	Simulate basic ALU operations (addition and subtraction).	04
10	Simulate basic ALU operations (addition and subtraction).	04
11	Create a basic HTML document with appropriate structure(DOCTYPE, html, head, body).	04
12	Use HTML tags to create headings (h1-h6) and paragraphs.	04
13	Add line breaks and horizontal rules to separate content on webpage.	04
14	Insert images into your HTML document using the img tag.	04

15	Create hyperlinks (both absolute and relative paths) to link to other web pages.	04
16	Design a table that displays student information: Name, Roll Number, and Grade for at least 5 students..	04
17	Create ordered and unordered lists with list items.	04
18	Design a table with rows and columns, and apply colspan and row span attributes.	04
19	Create an HTML page that embeds a video (e.g., an MP4 file) and an audio clip (e.g., MP3 file) with control options.	04
20	Create a static website	04

Course outcomes: By the end of the Course Students will be able to

1. Understand the basic structure and syntax of HTML.
2. Use standard HTML tags to design simple web pages.
3. Perform binary arithmetic and understand number systems.
4. Understand memory hierarchy and input/output operations.

References

1.Jon Duckett (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.

2. Jennifer Niederst Robbins (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics* (5th Edition). O'Reilly Media. Jennifer.

3..Guy Hart-Davis(2023) "The ABCs of Microsoft Office 97 Professional edition", BPB Publications.

4.. Karl Schwartz(1998), "Microsoft Windows 98 Training Guide" BPB Publications.

B. Sc. Part – II, Semester III Computer Science

Vocational Skill Course (VSC) I

BCSTVSC I: Object Oriented Programming using C++ (Practical Based Course)

(Total Credit-02)

Course objectives: By the end of the students will be able to:

1. Reinforce Theory with Hands-on Learning.
2. Develop Strong Coding Skills.
3. Learn Debugging and Error Handling.
4. Deep Understanding of Algorithms.

Practical No	Content	No. of hours per Practical
1	Write a C++ program to demonstrate the use of various data types and identifiers.	04
2	Create a C++ program to perform basic arithmetic operations.	04
3	Implement a C++ program to find the factorial of a given number.	04
4	Implement a C++ program to display the Fibonacci series up to a given limit using a do-while loop.	04
5	Develop a program to check whether a given number is a prime number or not using for loop.	04
6	Implement a program to display the pattern.	04
7	Implement a C++ program to demonstrate class and objects.	04
8	Write a C++ program for swapping two numbers.	04
9	Implement operator overloading in C++.	04
10	Write a C++ program to demonstrate the use of constructors (default, parameterized, copy) and destructors in a class.	04
11	Write a program to demonstrate the concept of inheritance with a base class "Shape" and derived classes (Rectangle, Circle) to calculate area and perimeter.	04
12	Develop a class "Employee" with static data members(employee count,	04

	total salary) and static member functions to display employee details and average salary.	
13	Write a C++ program to implement abstraction.	04
14	Write a C++ program to implement Polymorphism.	04
15	Implement a C++ program to read and write data to a file using file handling techniques.	04

Course outcomes: By the end of the Course students will be able to:

1. Gaining hands-on programming experience.
2. Improving problem-solving abilities.
3. Mastering C++ syntax and language features.
4. Developing debugging and testing skills, and gaining confidence in applying theoretical knowledge to real-world scenarios.

References

1. Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. Boston: Cengage Learning, 2014.
2. Koenig, Andrew, and Barbara E. Moo. *Accelerated C++: Practical Programming by Example*. Boston: Addison-Wesley, 2000.
3. Deitel, Paul, and Harvey Deitel. *C++ How to Program*. Boston: Pearson, 2017.
4. Prata, Stephen. *C++ Primer Plus*. Boston: Pearson Education, 2019.
5. Eckel, Bruce. *Thinking in C++*. Upper Saddle River: Prentice Hall, 2000.
6. Meyers, Scott. *Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14*. Sebastopol: O'Reilly Media, 2014.

B. Sc. Part – II, Semester III Computer Science

Skill Enhancement Course (SEC) I
BCST SEC I: System Analysis and Design
(Total Credit-02)

Course objectives: By the end of the Course Student will be able to

1. Broaden their knowledge of software engineering.
2. Learn Software testing algorithms and programs.
3. Gain skills in creating DFDs, ER diagrams and UML models.
4. Learn requirement gathering, analysis and documentation techniques

Section	Content	No. of hours per unit
I	**Introduction to System Concepts** 1.1 Definition and characteristics of a system 1.2 Types of systems 1.3 System development life cycle (SDLC) 1.4 Feasibility study – technical , operational and economic feasibility 1.5 Data flow diagrams (DFDs) 1.6 Data dictionary 1.7 Entity-Relationship diagrams (ERDs)	**08**
II	**System Design** 2.1 Design principles and objectives , Input design , Output design , User interface design and file and database design 2.2 Implementation strategies and Testing techniques and levels 2.3 Documentation – user manuals and system manuals , 2.4 Unified Modeling Language (UML) – use case, class ,activity and sequence diagrams.	**07**

Course outcomes: By the end of the Course Students will be able to…

1. Identify and define system requirements through detailed analysis and user feedback.
2. Apply structured and object-oriented methodologies for system development.
3. Understand the role of testing, validation, and documentation in ensuring successful system implementation

References

1. Awad, Elias M. *System Analysis and Design*. Boston: Cengage Learning, 2012.

2. Dennis, Alan, Barbara Haley Wixom, and Roberta Roth. *Systems Analysis and Design*. Hoboken: John Wiley & Sons, 2015.

3. Kendall, Kenneth E., and Julie E. Kendall. *Systems Analysis and Design*. Boston: Pearson, 2013.

4. Whitten, Jeffrey, and Lonnie Bentley. *Systems Analysis and Design Methods*. New York: McGraw-Hill Education, 2007.

5. Hoffer, Jeffrey A., Joey F. George, and Joseph S. Valacich. *Modern Systems Analysis and Design*. Boston: Pearson, 2016.

B. Sc. Part – II, Semester III Computer Science

Skill Enhancement Course (SEC) I

BCSPSEC 2: System Analysis and Design

(Total Credit-02)

Course objectives: By the end of the Student will be able to..

1. To develop the ability to analyze user requirements for system development.
2. To design system models using tools like DFD and UML diagrams.
3. To apply structured and object-oriented approaches in system design.
4. To understand and practice system documentation and reporting.

Practical No.	Content	No. of hours per Practical
1	Preparing Software Requirements Specifications.	04
2	Identifying Domain Classes from the Problem Statements.	04
3	Modeling UML Class Diagrams and Sequence diagrams	04
4	Modeling UML Use Case Diagrams and Capturing Use Case Scenarios.	04
5	E-R Modeling and State chart and Activity Modeling.	04
6	Modeling Data Flow Diagrams	04
7	Estimating size, effort, cost, time, and resources for a project.	04
8	Estimation of Test Coverage Metrics and Structural Complexity.	04
9	Conduct unit tests for the implemented system modules.	04
10	Create a detailed SRS document including functional and non-functional requirements.	04
11	Create an interface to connect and enable communication between two different systems.	04
12	Draw a context-level (Level 0) DFD and add more detail and draw a level1 DFD for your system.	04

13	Draw an Entity Relationship Diagram using basic symbols.	04
14	Design a sample login or data entry form and also design a simple report format (Input and Output design)	04
15	Create a small user manual with steps to use the system.	04

Course outcomes: By the end of the Course students will be able to…

1. Apply real-world problem-solving skills by modeling user and system interactions.
2. Translate business requirements into technical specifications through system design tools.
3. Create and interpret Data Flow Diagrams (DFD) and UML diagrams for system analysis.
4. Develop clear, precise system documentation to support maintenance and future development.

References

1. Kendall, Kenneth E., and Julie E. Kendall. *Systems Analysis and Design*. Boston: Pearson Education, 2013.
2. Dennis, Alan, Barbara Haley Wixom, and Roberta M. Roth. *Systems Analysis and Design*. Hoboken: John Wiley & Sons, 2015.
3. Hoffer, Jeffrey A., Joey F. George, and Joseph S. Valacich. *Modern Systems Analysis and Design*. Boston: Pearson, 2016.
4. George, Joey, Kevin C. Dittman, and Alan Dennis. *Object-Oriented Systems Analysis and Design*. Boston: Cengage Learning, 2004.

B. Sc. Part – II, Semester III Computer Science

Indian Knowledge System (IKS) 2

BCSTIKS 2: History of Computers in India

(Total Credit-02)

Course objectives: By the end of the Course students will be able to:

1. Understand design for a steam-powered, mechanical computer.
2. Learn digital computing replaced analog methods.
3. Study the evolution of Indian programming languages.
4. Identify the story behind the modern computing.

Unit	Content	No. of hours per unit
I	**Pre-Independence Era of Computers** 1.1 Introduction: Computing in the Pre-Independence world 1.2 Establishment of the TATA Institute of Fundamental Research 1.3 Analog Computing in the 19th and early 20th 1.4 Introduction of electronic computers in India 1.5 Information Technology before 1945.	**07**
II	**Early Computing Initiatives** 2.1 Development of the first indigenous electronics computer:HEC-2M, 2.2 Role of F.C.kohli in setting up the first computer manufacturing company in India 2.3 Computers and culture in the 1960s 2.4 Early Computer languages and software.	**07**
	Era of Mainframes and Minicomputers 3.1 Introduction of mainframe and minicomputer	

III	technologies in India, 3.2 Role of Department of Electronics(DoE) in promoting computerization, 3.3 Evolution of Indian programming languages(FORTRAN,COBOL)	**08**
IV	**Modern Computing** 4.1Emergence of Indian IT companies and software exports, 4.2 Role of NRIs in the growth of Indian IT industry, Internet and Digital Revolution, 4.3 Mobile Computing and smartphone Revolution in India, 4.4 Artificial Intelligence(AI) and Machine Learning (ML) in Indian industries	**08**

Course Outcomes:-Student will be able to….

1. Design for a steam-powered mechanical computer.
2. Evaluate digital computing replaced analog methods.
3. Gain the knowledge about evolution of Indian programming languages
4. Elaborate the story behind modern computing.

References

1. Campbell-Kelly, Martin, and William Aspray. *"Computer: A History of the Information Machine".* Boulder: Westview Press, 2004
2. Turner, Fred. *"From Counterculture to Cyberculture".* Chicago: University of Chicago Press, 2006
3. Campbell-Kelly, Martin. *"From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry".* Cambridge: MIT Press, 2003
4. Abbate, Janet. *"Inventing the Internet".* Cambridge: MIT Press, 1999

B. Sc. Part – II, Semester IV Computer Science

BCST 241: Object Oriented Concepts Using JAVA

(Total Credit-02)

Course objectives: By the end of the Course Student will be able to

1. Improve the analytical skills of object-oriented programming and formal introduction to Java programming language.
2. Understand Object Oriented Programming language.
3. Study abnormal termination of a program using exception handling.
4. Identify the principles of the applets and its GUI programming.

Unit	Content	No. of hours per unit
I	**Introduction To Java, Objects and Classes** 1.1 Introduction to object-oriented programming. 1.2 Basic concepts of OOP (Object, class, inheritance, polymorphism etc.) Advantages of OOP over Procedure oriented programming 1.3 History and features of Java Programming, Java Environment, Java tokens, constants, variables, data types, type casting, Operators and Expressions 1.4 Implementing Java Program, Branching and looping statements, Class, objects, methods, Constructors and destructor	**09**
II	**Inheritance, Polymorphism and Packages** 2.1 Defining sub class, subclass constructor, Inheritance- Multiple and hierarchical. 2.2 Defining packages, system packages, Creating & accessing packages, Adding a class to package. 2.3 Polymorphism- function overloading and over ridding 2.4 Difference between method overloading and overriding.	**07**
III	**Multithreading and Exception Handling** 3.1 Concept of thread, Life cycle of thread, Creating threads, extending a thread class- declaring the class, run() method, Stopping and blocking threads, Using thread method, Thread priority 3.2 Introduction to exception, Syntax of exception handling	**08**

	code, Multiple catch statement, Using finally statement, Throwing exception, user defined exception.	
IV	**AWT and Event Handling** 4.1 Introduction to Abstract Window Toolkit (AWT) 4.2 Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames 4.3 Layout Managers: Flow Layout, Border Layout, Grid Layout	**06**

Course outcomes - Students should be able to

1. Demonstrate professionally acceptable coding and performance standard.
2. Learn the basic principles of the object-oriented programming.
3. Demonstrate an introductory understanding of graphical user interfaces, multithreaded programming, and event-driven programming.
4. Identify the principles of the applets and its GUI programming.

References

1. Schildt, Herbert. *Complete Reference Java*. New York: McGraw-Hill Education, 2002.
2. Holzner, Steven. *Java 2 Programming Black Book*. Scottsdale: Paraglyph Incorporated, 2001.
3. Balagurusamy, E. *Programming with Java: A Primer*. New Delhi: McGraw-Hill Education, 2010.
4. Horstmann, Cay S., and Gary Cornell. *Core Java Volume I – Fundamentals*. Upper Saddle River: Prentice Hall, 2007.
5. Salokhe, Rajendra. *Java Programming*. Pune: Aruta Publication, 2008.

B. Sc. Part – II, Semester IV Computer Science

BCST 242: Operating Systems

(Total Credit-02)

Course objectives: By the end of the Course Students will be able to

1. Imbibe brief about OS organization
2. Study memory management techniques.
3. Learn Shell operating system.
4. Understand the basic organization of operating system.

Unit	Content	No. of hours per unit
I	**Fundamental Concepts** 1.1 System Software, Resource Abstraction, OS strategies. 1.2 Types of operating systems –Windows, Linux/Ubuntu 1.3 Multiprogramming, Batch, Time Sharing, Single user and Multiuser, 1.4 Process Control & Real Time Systems.	**08**
II	**Operating System Organization** 2.1 Factors in operating system design, basic OS functions, implementation consideration. 2.2 Process modes, methods of requesting system services – system calls and system programs.	**07**
III	**Process Management and Memory Management** 3.1 System view of the process and resources, initiating the OS, process address space, process abstraction, resource abstraction, process hierarchy, Thread model 3.2 Scheduling Mechanisms, Strategy selection, non-pre-emptive and preemptive strategies. 3.3 Mapping address space to memory space, memory allocation strategies, fixed partition, variable partition, paging, virtual memory.	**08**
IV	**Shell introduction and Shell Scripting** 4.1 What is shell and various type of shell, Various editors present in linux 4.2 Different modes of operation in vi editor ,What is shell script, Writing and executing the shell script ,Shell variable (user defined and system variables) 4.3 System calls, Using system calls, Pipes and Filters 4.4 Decision making in Shell Scripts (If else, switch), Loops in shell, Functions ,Utility programs (cut, paste, join, tr , uniq utilities) ,Pattern matching utility (grep).	**07**

Course outcomes: By the end of the Course Student should able to

1. Learn the fundamentals of operating systems and its types.
2. Imbibe the basic Operating System Organization.
3. Identify the process & memory management in Operating System.
4. Design programs using a Shell Scripting

References

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. *Operating System Concepts*. Hoboken: John Wiley & Sons, 2008.

2. Tanenbaum, Andrew S. *Modern Operating Systems*. Boston: Pearson Education, 2007.

3. Nutt, Gary. *Operating Systems: A Modern Perspective*. Boston: Pearson Education, 2003.

4. Stallings, William. *Operating Systems: Internals and Design Principles*. New Delhi: Prentice Hall of India, 2008.

5. Milinkovic, M. *Operating Systems: Concepts and Design*. New Delhi: Tata McGraw-Hill, 2009.

B. Sc. Part – II, Semester IV Computer Science

BCSP243: Practicals based on BCST 241: Object Oriented Concepts Using JAVA and BCST 242: Operating Systems

Course objectives: By the end of the students should be able to…

1. Improve the analytical skills of object-oriented programming and formal introduction to Java programming language.
2. Understand Object Oriented Programming language.
3. Study abnormal termination of a program using exception handling.
4. Understand Shell operating system and memory management techniques.

Practical No	Content	No. of hours per practical
1	Program to define a structure of a basic JAVA program.	04
2	Program to define the data types, variable and operator.	04
3	Program to define the arrays and decision making and loop statements.	04
4	Program to define the Jump statements and type casting.	04
5	Program to define class and constructors. Demonstrate constructors.	04
6	Program to define class and inheritance. Demonstrate simple inheritance.	04
7	Program to define class and inheritance. Demonstrate multilevel inheritance.	04
8	Program to define class and inheritance. Demonstrate hierarchical inheritance.	04
9	Program to define class, methods, and objects. Demonstrate method overloading.	04
10	Program to define inheritance and show method overriding, demonstrate user define Packages and API Packages, demonstrate Applet structure and event handling	04
11	Write a program to check status of keyboard using interrupt handler	04
12	Write a program to implement copy command of DOS ,Program to display date and time of system	04
13	Write a program to implement pwd and wc command of linux.	04
14	Usage of following commands: ls, pwd, tty, cat, who, who am I, rm,	04

	mkdir, rmdir, touch, cd.	
15	Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.	04
16	Usage of following commands: chmod, grep, tput (clear, highlight), bc.	04
17	Write a shell script to modify "cal" command to display calendars of the specified range of months	04
18	Write a shell script to accept a login name. If not a valid login name display message – "Entered login name is invalid" and shell script to display date in the mm/dd/yy format.	04
19	Write a shell script to display on the screen sorted output of "who" command along with the total number of users.	04
20	Case study on Different Modes of Operation in Vi Editor, Shell Script, Writing.	

Course outcomes: By the end of the Course Students will be able to…

1. Derive the process & memory management in Operating System and the basicOperating System organization.
2. Analyze professionally acceptable coding and performance standard.
3. Demonstrate graphical user interfaces, multithreaded programming, and event-driven programming.
4. Learn the basic principles of the object-oriented programming

References

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. *Operating System Concepts*. Hoboken: John Wiley & Sons, 2008.
2. Tanenbaum, Andrew S. *Modern Operating Systems*. Boston: Pearson Education, 2007.
3. Nutt, Gary. *Operating Systems: A Modern Perspective*. Boston: Pearson Education, 2003.
4. Stallings, William. *Operating Systems: Internals and Design Principles*. New Delhi: Prentice Hall of India, 2008.
5. Milinkovic, M. *Operating Systems: Concepts and Design*. New Delhi: Tata McGraw-Hill, 2009.
6. Dhamdhere, D. M. *System Programming and Operating System*. New Delhi: Tata McGraw-Hill, 2002.
7. Kanetkar, Yashwant. *Unix Shell Programming*. New Delhi: BPB Publications, 2003.
8. Schildt, Herbert. *Complete Reference Java*. New York: McGraw-Hill Education, 2002.

9. Holzner, Steven. *Java 2 Programming Black Book*. Scottsdale: Paraglyph Incorporated, 2001.

B. Sc. Part – II, Semester IV Computer Science

BCST 244: Cyber Security

(Total Credit-02)

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

1. To Identify and comprehend various types of cyber threats and attacks.
2. To Familiarize with cyber laws, regulations, and ethical considerations.
3. To examine network security, congestion control, and wireless networking principles.
4. To Introduce students to various types of cyber threats and attacks.

Unit	Content	No. of hours per unit
I	**Fundamentals of Cyber security** 1.1 Basics of Cyber security 1.2 Importance of Cyber security,Cyber security Fundamentals 1.3 Types of Cyber Attacks 1.4 security vulnerabilities, threats and attacks 1.5 Types of Active attacks, Types of Passive attacks	**08**
II	**Cyberspace and the Law & Cyber Forensics** 2.1 Introduction, Cyber Security Regulations 2.2 The INDIAN Cyberspace 2.3 National Cyber Security Policy 2.4 Digital Forensics Science 2.5 Forensics Analysis of Email, Digital Forensics Lifecycle. 2.6 Challenges in Computer Forensics	**07**
III	**Database Security** 3.1 Introduction to Database Security 3.2 Authentication and Authorization 3.3 Database Encryption 3.4 Auditing and Monitoring 3.5 Database Security in Cloud Environments .	**07**
IV	**Internet and Web Application Security** 4.1 Web Security: Web authentication, Injection Flaws, SQL	**08**

	Injection. 4.2 Web Browser Security. 4.3 E-Commerce Security 4.4 Web Application Firewalls (WAF)	

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

1. Learn what cyber crime is and why it is important.
2. Understand basic networking and the OSI model
3. Learn about common network attacks.
4. Understand web application attacks and vulnerabilities.
5. Learn the basics of security operations and defense strategies.

References

1.Preston Gralla, “How Personal and Internet Security Work”, QuePublications, 2004

2. Alfred Basta and Wolf Halton, Computer Security Concepts, Issues and Implementation,Cengage Learning, 2020

3. Joseph Pelton , Indu B.Singh, “Digital Defense: A Cybersecurity Primer”, Copernicus, 2025.

4. John R. Vacca, “Computer and Information Security Handbook”, Morgan Kaufmann, 3rd Edition, 2027

5. Brian Craig, “Cyberlaw: The Law of the Internet and Information Technology”, Lexis Nexis publishing, 2024

6. Jason Andress, Steve Winterfeld, “Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners2nd Edition “, Syngress publishing, 2023

B. Sc. Part – II, Semester IV Computer Science

BCST 245: Design Principle using CSS

(Total Credit-02)

Course objectives: By the end of the Course Students will be able to

1. Understand core CSS layout and styling concepts.
2. Apply design principles using CSS.
3. Create responsive and accessible web designs
4. Create dynamic websites.

UNIT	Content	No. of hours per unit
I	**Introduction to CSS and Web Design** 1.1 History and Evolution of CSS, Role of CSS in Web Development. 1.2. Basic Syntax and Structure of CSS, Integrating CSS with HTML. 1.3. Inspecting and Editing CSS in Browsers. 1.4 Adding CSS in HTML.	**08**
II	**Selectors** 2.1 Introduction of selectors 2.2Types of Selectors. 2.3 CSS Elements Used in Web Design 2.4 Web Styling with CSS	**07**
III	**CSS Box Model** 3.1 Introduction to Box model. 3.2 Images and Background. 3.3Display and Position, Colors, Fonts, lists and Tables. 3.4 Pseudo-classes and Combinators, Debugging and Dev tools, Flexbox	**07**

	and Grid.	
IV	**CSS Flexbox And Grid** 4.1 Introduction, Flex Container, Flex Items, Responsive Flexbox. 4.2 Grid Introduction. 4.3 Grid Columns, Rows and Gaps 4.4. Grid Container, Grid Item.	**08**

Course outcomes: By the end of the Course Students will be able to**:**

1. Understand the role of CSS in separating structure and presentation in web design.
2. Identify and apply appropriate CSS selectors and properties to style HTML elements in alignment with design principles.
3. Construct aesthetically pleasing and user-friendly web pages using CSS box model, and layout techniques.
4. Develop responsive designs that adapt to various screen sizes using CSS media queries and flexible units.

References

1. **Jon Duckett**,*HTML and CSS: Design and Build Websites*,Publisher: *Wiley, 2011*.
2. **Eric A. Meyer**,*CSS: The Definitive Guide*, 4th Edition,**P**ublisher : O'Reilly Media, 2017.
3. Ben Frain, *Responsive Web Design with HTML5 and CSS*, 4th Edition, Publisher**:** Packt Publishing, 2022.
4. Thomas Powell, *HTML & CSS: The Complete Reference*, 5th Edition**,**Publisher**:** McGraw-Hill Education, 2010.
5. Chris Coyier, *CSS Secrets: Better Solutions to Everyday Web Design Problems*, **Publisher:** O'Reilly Media, 2015.
6. Rachel Andrew, *The New CSS Layout*, Publisher**:** A Book Apart, 2017.

B. Sc. Part – II, Semester IV Computer Science

BCSP 246: Practicals based on BCST244: Cyber Security

and

BCST245: Design Principles using CSS

(Total Credits-02)

Course objectives: By the end of the Course Students will be able to to…

1. To Identify and comprehend various types of cyber threats and attacks.
2. To Familiarize with cyber laws, regulations, and ethical considerations.
3. Create responsive and accessible web designs
4. Create dynamic websites.

Practical No	Content	No.of hours per practical
1	Study of the features of firewall in providing network security and to set Firewall Security in windows.	04
2	Steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome)	04
3	Study of different types of vulnerabilities for hacking a websites / Web Applications.	04
4	Case Study On basic technology in cyber security.	04
5	Analysis the security vulnerabilities of Email application	04
6	Perform encryption and decryption of Caesar cipher Write a script for performing these operations	04
7	Case Study on – Cyber Law	04
8	Case Study on – Cyber Harassment	04
9	Case Study on – Patent Law.	04
10	Case Study on – Confidentiality, integrity, and availability (CIA triad)	04
11	Write a program on basic CSS Styling.	04
12	Inspecting and Editing CSS	04

13	Build structured layouts with CSS Grid.	04
14	Apply CSS rules to specific HTML elements using various selectors.	04
15	Design a simple webpage layout with CSS.	04
16	Apply and analyze the CSS box model properties.	04
17	Integrate and style images and backgrounds using CSS.	04
18	Control element layout with CSS display and positioning.	04
19	Create flexible layouts with CSS Flexbox.	04
20	Build structured layouts with CSS Grid.	04

Course outcomes: By the end of the Course Students will be able to…

1. Learn about common network attacks.
2. Understand web application attacks and vulnerabilities.
3. Construct aesthetically pleasing and user-friendly web pages using CSS box model, and layout techniques.
4. Develop responsive designs that adapt to various screen sizes using CSS media queries and flexible units.

References

1. **Jon Duckett**,*HTML and CSS: Design and Build Websites*,**Publisher:** *Wiley, 2011*.
2. **Eric A. Meyer**,*CSS: The Definitive Guide*, 4th Edition,**Publisher** : O'Reilly Media, 2017.
3. Ben Frain, *Responsive Web Design with HTML5 and CSS*, 4th Edition, **Publisher:** Packt Publishing, 2022.
4. Thomas Powell, *HTML & CSS: The Complete Reference*, 5th Edition,**Publisher:** McGraw-Hill Education, 2010.
5. Chris Coyier, *CSS Secrets: Better Solutions to Everyday Web Design Problems*, **Publisher:** O'Reilly Media, 2015.
6. Rachel Andrew, *The New CSS Layout*, **Publisher:** A Book Apart, 2017.

B. Sc. Part – II, Semester IV Computer Science

Vocational Skill Course (VSC) II

BCSTVSC II Vocational Skill Course: Photoshop Designer

Course objectives: By the end of the Course Student should be able to…

1. learn and applying Principals for photoshop.
2. Study the ways to balance between formal theories with practical applications.
3. Identify deep knowledge of Adobe photoshop cs2.
4. Understand the fundamental concepts of photo editing.

Practical No.	Content	No. of hours per Practical
1	Study of the graphics design software's, its types and uses	04
2	Understand the menus of software of Adobe.	04
3	Draw the elements of design: (line, shape, Forms, space, colors etc.)	04
4	Study of the photoshop software's	04
5	Browse the image and resize the image in Adobe Photoshop	04
6	Perform the properties of image in photoshop.	04
7	Design the images.	04
8	Redesign an existing image.	04
9	Drawing and sketching the image and logo.	04
10	Sketching of natural and manmade objects and environment.	04
11	Sketching of representational Drawing	04
12	Draw the image and use color Correction property in it.	04
13	Understand the features and 3D effects	04
14	Create a Poster or Flyer Design in Photoshop	04
15	Design a Creative Typographic Quote Poster	04

Course outcomes: By the end of the Course students will be able to…

1 Demonstrate in between drawing and execution with squash and stretch, staging, arc, principle.

2 Create organize content and sequences for photo editing.

3 Evaluate a thorough understanding of the elements of photoshop design.

4 Create advanced editing techniques for documentary

Reference

1. Chavez, Conrad, and Andrew Faulkner. *"Adobe Photoshop Classroom in a Book (2024 Release)".* San Francisco: Adobe Press, 2024.
2. Joyner, Joseph. *"Photoshop CC for Beginners: The Ultimate Digital Photography and Photo Editing Tips and Tricks Guide for Creating Amazing Photos".* [Place of publication not specified]: Mihails Konoplovs, 2015.
3. Kelby, Scott. *"The Adobe Photoshop Book for Digital Photographers".* San Francisco: New Riders, 2023.
4. Barker, Corey. *"Photoshop Down & Dirty Tricks for Designers".* Berkeley: New Riders, 2011.
5. Laskevitch, Stephen. *"Adobe Photoshop: A Complete Course and Compendium of Features".* San Rafael: Rocky Nook, 2020.

B. Sc. Part – II, Semester IV Computer Science

Skill Enhancement Course (SEC) II

Skill Enhancement Course BCSTSEC 2: XML Programming

Course objectives: By the end of the Course Student should be able to…

1. Learn and demonstrate their understanding and skillful use of the elements and principles of XML
2. Understand skill to use the digital tools as a powerful means of communication for creation, modification & presentation.
3. Identify aesthetic sensibilities into their works and explore ways to balance between formal theories with practical applications.
4. To use XML with different language

Unit	Content	No. of hours per unit
I	**Introduction to XML** 1.1 Introduction to Markup Languages 1.2 Evolution of SGML and HTML 1.3 Need for XML 1.4 Applications of XML 1.5 Creating XML Documents, DOCTYPE Definition, Document 1.6 Type Definition (DTD), Document Type Definition (DTD) 1.7 Comments, Creating Elements and Attributes.	**08**
II	**XML Style Sheets** 2.1 The Use of Style Sheets, Using Cascading Style Sheets (CSS), XSL - The Style Sheet for XML Documents, XSL Style Sheet Template, Hyperlinks in XML Documents, Links and DTD 2.2 XML Document Object Model, Basic Features of the Document Object Model, XML Object Interfaces, Components of an XML DOM Tree Structure 2.3 XML Query Language	**07**

Course outcomes: By the end of the Course Students will be able to…

1. Learn the basics of creating, transforming, and validating XML documents.

2. Understand XQuery fundamentals, its uses.

3. How to run complex queries on XML data.

4.How to use XML with other languages

References

1. Fawcett, Joe, Danny Ayers, and Liam R. E. Quin. *"Beginning XML".* Indianapolis: Wrox Press, 2012.
2. Ray, Erik T. *"Learning XML: Creating Self-Describing Data".* Sebastopol: O'Reilly Media, 2003.
3. Harold, Elliotte Rusty, and W. Scott Means. *"XML in a Nutshell: A Desktop Quick Reference".*Sebastopol: O'Reilly Media, 2004.
4. Evjen, Bill, et al. *"Professional XML".* Indianapolis: Wrox Press, 2007.
5. Hoque, Reaz. *"XML for Real Programmers".* San Diego: Morgan Kaufmann, 2000.

B. Sc. Part – II, Semester IV Computer Science

Skill Enhancement Course (SEC) II

Skill Enhancement Course BCSTSEC 2: XML Programming

Course objectives: By the end of the Course Student should be able to…

1. Learn all the essential elements of programming
2. Explore all the main XML techniques
3. Learn to present raw data attractively and efficiently in browsers using XSL style sheets
4. Discover how to reuse basic code to create any kind of XML data management program

Practical No.	Content	No. of hours per Practical
1	Creating a simple XML document.	04
2	Creating a simple XML document with attributes	04
3	Create an XML file using the Internal DTD.	04
4	Create an external DTD and implement it in XML file.	04
5	Creating a XML Schema.	04
6	Create an xml file to implement CSS concept.	04
7	To create a simple XSLT transformation from XSL to XML.	04
8	Create Two or more XML namespaces for different categories	04
9	Create XML code to student profile with optional and nested elements.	04
10	Create a XML file to implement XPATH concept.	04
11	Creating a XML DOM class.	04
12	Create an XML document for student information and college course listing.	04
13	Create a well-formed XML document containing details of a car like: id, company name, model, engine and mileage	04
14	Create a valid XML document containing details of a car like: id, company name, model, engine and mileage using DTD.	04

15	Write a program to connect a XML web page to any database engine.	04

Course outcomes: By the end of the Course students will be able to…

1. Learn how to create an XML Schema.
2. Learn how to verify that an XML document is valid according to its Schema.
3. Develop a Program using XML.
4. Integrate knowledge and skills to develop XML programming.

References

1. Fawcett, Joe, Danny Ayers, and Liam R. E. Quin. *"Beginning XML".* Indianapolis: Wrox Press, 2012.
2. Ray, Erik T. *"Learning XML: Creating Self-Describing Data".* Sebastopol: O'Reilly Media, 2003.
3. Harold, Elliotte Rusty, and W. Scott Means. *"XML in a Nutshell: A Desktop Quick Reference".*Sebastopol: O'Reilly Media, 2004.
4. Evjen, Bill, et al. *"Professional XML".* Indianapolis: Wrox Press, 2007.
5. Hoque, Reaz. *"XML for Real Programmers".* San Diego: Morgan Kaufmann, 2000.