Rayat Shikshan Sanstha’s

Yashavantrao Chavan Institute of Science, Satara (Autonomous)

Lead College, Karmaveer Bhaurao Patil University

Reaccredited by NAAC (3rdCycle) with 'A+' grade (CGPA 3.57).

ISO 9001:2015 Certified

Bachelor of Science

Part - II

Computer Science

Syllabus

to be implemented w. e. f. June, 2024

NEP 2020

Rayat Shikshan Sanstha's
Yashavantrao Chavan Institute of Science, Satara
Department of Computer Science
Syllabus for Bachelor of Science (Computer Science) Part II

1. **TITLE:** COMPUTER SCIENCE

2. **YEAR OF IMPLEMENTATION:** New Syllabi for the B.Sc. II Computer Science will be implemented from June 2024 onwards.

3. **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.

4. **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.

5. **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.

6. **DURATION:**
    - The course shall be a full-time course.

7. **PATTERN:** Semester Examination

8. **MEDIUM OF INSTRUCTION:** English

Course Structure B. Sc II Semester III

Sr. No.	Course Category	Course Code	Name of Course
1	Major I	BCST 231	Data Structure using C
2	Major II	BCST 232	Computational Mathematics
3	Major Practical I	BCSP 233	Based on BCST 231
4	Major Practical II	BCSP 234	Based on BCST 232
5	Minor-I	BCST 235	Basics of HTML
6	Minor Practical-I	BCSP 236	Based on BCST 235
7	VSC	BCSTVSC 1	Object Oriented Programming using C++
8	SEC	BCSTSEC 2	System Analysis And Design
9	AEC	BCSTAEC 1	English I
		BCSTAEC 2	English II
10	VEC	BCSTVEC 1	Environmental Awareness for Computer Science

Course Structure B. Sc II Semester IV

Sr. No.	Course Category	Course Code	Name of Course
1	Major I	BCST241	Object Oriented Concept using Java
2	Major II	BCST242	Computational Statistics
3	Major Practical I	BCSP243	Based on BCST 241
4	Major Practical II	BCSP244	Based on BCST 242
5	Minor-I	BCST245	Operating System
6	Minor Practical-I	BCSP246	Based on BCST 245
7	VSC	BCSTVSC2	Fundamental of Cyber Security
8	SEC	BCSTSEC3	XML Programming
9	AEC	BCSTAEC 1	English III
		BCSTAEC 2	English IV
10	CC	BCSTCC2	Fundamental of Physical Education

Major Syllabus

Course Structure for B.Sc. II (Semester- III)

Theory				Practical				
Course Title	**Course Code**	**Lecture per week**	**Credits**	**Course**	**Course Title**	**Course Code**	**Lecture per week**	**Credits**
Data Structure using C	BCST 231	5	2	Practical -1	Major Practical- I	BCSP 233	4	2
Computational Mathematics	BCST 232		2	Practical – 2	Major Practical- II	BCSP 234		

Note: B: B. Sc. T=Theory and P= Practical

Structure and Title of Courses of B. Sc. Course:

* B. Sc. II Semester III *

Course Number	Course Code	Course Name
V	BCST 231	Data Structure using C
VI	BCST 232	Computational Mathematics
Lab-I	BCSP 233	Major Practical- I
Lab-II	BCSP 234	Major Practical- II

B.sc II -Semester-III
Theory: Course-VI BCST 231: Data Structure using C

Course Objectives: 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 andanalyze various algorithms.

Credits =2	**SEMESTER-III** **BCST 231** **Data Structure using C**	**No. of hours per unit/credits**
UNIT - I	**Basic of Data Structures and Algorithms C**	**(8)**
	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).	
UNIT - II	**Stack**	**(7)**
	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.	
UNIT - III	**Queue**	**(7)**
	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	
UNIT - IV	**Linked List**	**(8)**
	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 intolinked list, Deletion from a linked list.	

Course outcomes - Students should be able to

1. Learn the fundamentals of c and ability to choose appropriate data structures torepresent data items in real world problems.
2. Analyze the different types of stack notations.
3. Design programs using a variety of data structures such as stacks, queues.
4. Demonstrate various kinds of linked list.

Reference Books:

1. "Data Structures and Algorithms in C" by Michael T. Goodrich, Roberto Tamassia, and David M. Mount (Year: 2018)
2. "Data Structures Using C" by E. Balagurusamy (Year: 2017)
3. "C Programming Absolute Beginner's Guide" by Perry, Greg (Year: 2016)
4. "Problem Solving and Program Design in C" by Hanly, Jeri R. (Year: 2015)
5. "Modern Data Structures Using C" by R. S. Salaria (Year: 2017)
6. "Mastering Algorithms with C" by Kyle Loudon (Year: 1999, but updated edition in 2018)
7. "Data Structures and Algorithmic Thinking with Python" by Narasimha Karumanchi (Year: 2015)
8. "Data Structures: A Pseudocode Approach with C" by Richard F. Gilberg and Behrouz A. Forouzan (Year: 2015)

BCSP 233: Lab Course (Data Structures using C)

Course Objectives: Student 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.

Credits=2	**Lab Course (Data Structure using C)**	**No. of hours per unit/credits**
	Data Structure using C	**(30)**
	Exercise No.1. Write a linear searching algorithm with example. **Exercise No.2** Write a binary searching algorithm with example. **Exercise No.3.** Write a Bubble sort algorithm with example. **Exercise No.4.** Write an Insertion sort algorithm with example. **Exercise No.5.** Write a selection sort algorithm with example. **Exercise No.6.** Write Pseudo code algorithm to find summation of given n numbers. **Exercise No.7.** Problems on Big O notations. **Exercise No.8**. Problems on omega notations. **Exercise No.9.** Problems on Theta notations. **Exercise No.10**. STACK Implementation using Array with PUSH, POP, and display Operations. **Exercise No.11.** STACK Implementation using Link list with PUSH, POP, and display Operations. **Exercise No.12.** Perform Queues operations using simple Array implementation. Use Templates. **Exercise No.13.** Perform Queues operations using link list implementation. Use Templates. **Exercise No.14**. Create and perform different operations on Double-ended Queues using array implementation. **Exercise No.15**. Create and perform different operations on Double-ended Queues using Linked List implementation.	

	Exercise No.16 Perform Queues operations on circular queue using array implementation. **Exercise No.17.** Perform Queues operations on circular queue using link list implementation. **Exercise No.18.** Perform Queues operations on priority queue using array implementation. **Exercise No.19.** STACK Implementation with Linked List using C Program. **Exercise No.20.** Linked List Implementation using C Program.	

Course outcomes-Students should be able to

1. Prepare and perform an installation of Windows Server 2008 and identify the various typesof file Systems and their components.
2. Identify Active Directory logical components and infrastructure, create and manage fileSystem access security.
3. Learn the fundamentals of c and ability to choose appropriate data structures torepresent data items in real world problems.
4. Analyze the time and space complexities of algorithms.

Reference Books:

1. "Data Structures and Algorithms in C" by Michael T. Goodrich, Roberto Tamassia, and David M. Mount (Year: 2018)
2. "Data Structures Using C" by E. Balagurusamy (Year: 2017)
3. "C Programming Absolute Beginner's Guide" by Perry, Greg (Year: 2016)
4. "Problem Solving and Program Design in C" by Hanly, Jeri R. (Year: 2015)
5. "Modern Data Structures Using C" by R. S. Salaria (Year: 2017)
6. "Mastering Algorithms with C" by Kyle Loudon (Year: 1999, but updated edition in 2018)
7. "Data Structures and Algorithmic Thinking with Python" by Narasimha Karumanchi (Year: 2015)
8. "Data Structures: A Pseudocode Approach with C" by Richard F. Gilberg and Behrouz A. Forouzan (Year: 2015)

Syllabus – Major II

B. Sc. II – Semester – III

Theory: BCST 232: Computational Mathematics I

Course Objectives: Students will be able to

1. Understand types of Matrices and their applications, develop the skills find the Eigen values and Eigen vectors.
2. Impart adequate knowledge concept of graphs, and apply graph algorithms.
3. Choose and apply appropriate numerical techniques.
4. Solve the problem and interpret the result.

Credits=4	**SEMESTER – IV** **BCST 232: Computational Mathematics I**	**No. of hours per unit/ credits**
Credit –I **UNIT I**	**UNIT I: Matrices**	(10)
	1.1 Introduction of matrices. 1.2 Operations on Matrices and examples. 1.3 Definition: Inverse of Matrices, Examples on Inverse of Matrices by adjoint method, Rank of a matrix. 1.4 Solution of simultaneous linear equations: Homogeneous and Non-Homogeneous. 1.5 Eigen values and Eigen Vectors.	
Credit –I **UNIT II**	**UNIT II: Graph Theory**	(7)
	2.1 Definition and elementary results. 2.2 Types of graphs, Isomorphism, Matrix representation of graphs: Adjacency matrix and incidence matrix. 2.3 Definitions: walk, trail, tour, path and circuit, Definitions of connected, disconnected graphs 2.4 Dijkstra's shortest path algorithm. 2.5 Tree: Definition, Kruskal's algorithm.	
Credit –I **UNIT III**	**UNIT III: Numerical Interpolation**	(7)
	3.1 Definition: Forward Operator, Backward Operator, Shift Operator. 3.2 Central Difference. 3.3 Forward Difference Interpolation, Backward Difference Interpolation. 3.4 Divided Difference Interpolation. 3.5 Lagrange's Interpolation.	
Credit –I **UNIT IV**	**UNIT IV: Numerical Differentiation and Integration**	(6)
	4.1 Numerical Differentiation: Eulers Method 4.2 Runge-Kutta second order method, Runge-Kutta fourth order method 4.3 Numerical Integration-Trapezoidal Rule, Simpson's 1/3 rule, Simpsons 3/8 Rule, Weddle's Rules.	

Course Outcomes:

At the end of this course, students should be able to:

1. Solve the system of equations using the language of Matrices and elementary concepts of Matrices.
2. Analyze concepts of Graph and types of graphs and apply concepts of algorithms of graphs.
3. learn various methods of numerical differentiation and integration.
4. Choose appropriate numerical methods and determine the solutions to differential equations.

Reference Books:

1. Discrete Mathematics - Seymour Lipschutz, Marc Lars Lipson Schaum's Outlines- by, 3rd Edition., Tata McGraw Hill, Education Pvt. Ltd., New Delhi. 5th Reprint, 2012.
2. Discrete Mathematics and its applications, Seventh Edition, Kenneth. H. Rosen, Mc Graw Hill Publishing Company,2012.
3. Discrete Mathematics, M. Venkataraman, N. Sridharan and N. Chandrasekaran, The National Publishing Company,2009.
4. "Numerical Recipes: The Art of Scientific Computing" by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery (Year: 2019, 3rd Edition)
5. "Introduction to Numerical Analysis Using MATLAB" by Rizwan Butt (Year: 2018)
6. "Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics" by Justin Solomon (Year: 2020)

BCST 234: Lab Course (Computational Mathematics I)

Course Objectives: students will be able to:

1. Understand how matrices and determinants are used as mathematical tools in QA and determining the inverse of matrix.
2. Study the applications of graph algorithm and representation of mathematical expression in Polish notation.
3. Understand how numerical methods are used as mathematical tools and determining the results.
4. Study and implement C-programs of numerical integration methods.

Credits=2	**SEMESTER – III** **Major Practical II (BCST 234: Computational Mathematics I)**	**No. of hours per unit/ credits (30)**
	1. Solution of Homogeneous System of Linear Equation. 2. Solution of Non-Homogeneous System of Linear Equation. 3. Inverse of Matrix by Adjoint Method. 4. Verification of Cayley Hamilton Method. 5. Eigen Values and Eigen Vectors. 6. Isomorphism of Graph. 7. Matrix Representation of Graph. 8. Dijkstra’s shortest path Algorithm 9. Kruskal’s Algorithm. 10. Polish notations and Arborescence. 11. Newton Gregory Forward Interpolation. 12. Newton Gregory Backward Interpolation. 13. Newton’s Divided Difference Interpolation. 14. Lagrange’s Interpolation. 15. C-program to implement Euler Method. 16. C-program to implement Runge-Kutta 2nd Order Method. 17. C-program to implement Runge-Kutta 4th Order Method. 18. C-program to implement Trapezoidal Rule. 19. C-program to implement Simpson’s (1/3)rd Rule. 20. C-program to implement Simpson’s (3/8)th Rule.	

Course Outcomes: Students able to

1. Critically analyze and construct Mathematical arguments that relate to the study of introductory Matrix theory.
2. Analyze various graph algorithms and represent mathematical expression in Arborescence and Polish notation.
3. Learn various methods of numerical differentiation and integration.
4. Learn implementation of numerical methods through C-programming.

Reference Books:

1. Discrete Mathematics - Seymour Lipschutz, Marc Lars Lipson Schaum’s Outlines- by, 3rd Edition., Tata McGraw Hill, Education Pvt. Ltd., New Delhi. 5th Reprint, 2012.
2. Discrete Mathematics and its applications, Seventh Edition, Kenneth. H. Rosen, Mc Graw Hill Publishing Company,2012.
3. Discrete Mathematics, M. Venkataraman, N. Sridharan and N. Chandrasekaran, The National Publishing Company,2009.
4. "Numerical Recipes: The Art of Scientific Computing" by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery (Year: 2019, 3rd Edition)
5. "Introduction to Numerical Analysis Using MATLAB" by Rizwan Butt (Year: 2018)
6. "Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics" by Justin Solomon (Year: 2020)

Minor Syllabus

Course Structure for B.Sc. II (Semester- III)

Theory				Practical				
Course Title	**Course Code**	**Lecture per week**	**Credits**	**Course**	**Course Title**	**Course Code**	**Lecture per week**	**Credits**
Basics of HTML	BCST 235	2	2	Practical-1	Minor Practical -I	BCSP 236	2	2

Note: B: B. Sc. T=Theory and P= Practical

Structure and Title of Courses of B. Sc. Course:

* B. Sc. II Semester III *

Course Number	Course Code	Course Name
V	BCST 235	Basics of HTML
Lab-I	BCSP 236	Minor Practical -I

Minor -I

B.Sc. II– Semester – III

BCST 235: Basics of HTML

Course Objectives: Students will be able to

1. Understand HTML structure and syntax to create well-formed web documents.
2. Create structured and semantic web content using HTML elements for accessibility.
3. Implement hyperlinks, images, and multimedia effectively in web pages.
4. Design and implement basic web forms for user interaction and data submission.

Credits=2	**SEMESTER-I** **Paper V: Basics of HTML**	**No. of hours per unit/ credits**
UNIT I	**Unit – I: Introduction to HTML**	**(8)**
	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) 1.5 HTML Tags, formatting tags, Creating an HTML document 1.6 Applications of HTML 1.7 Features and Limitations Of HTML.	
UNIT II	**UNIT II - Exploring HTML Elements and Functions**	**(8)**
	2.1 Golden rules of web designing, Introduction to elements of HTML 2.2Working with Text, HTML Comments 2.3Working with Lists, Tables and Frames. 2.4 Formatting tables using colspan, rowspan, and CSS 2.5Working with Hyperlinks, Images and Multimedia 2.6 Working with Forms and controls, HTML Styles 2.7 Linking to external resources (images, CSS files, other web pages)	
UNIT III	**UNIT III: Introduction to Cascading Style Sheet**	**(7)**
	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.	
UNIT IV	**UNIT IV: CSS Box Model Fundamentals**	**(7)**
	4.1Box 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.	

Course Outcomes:

At the end of this course, students should be able to:

1. Students will be able to create structured and semantic HTML documents using appropriate tags and elements.
2. Students will learn how to format text, create hyperlinks, insert images, and embed multimedia content in HTML.
3. Participants will gain proficiency in designing and implementing basic web forms for data input and user interaction.
4. Students will develop the skills to optimize HTML code for accessibility, search engine optimization (SEO), and cross-browser compatibility.

Reference Books:

1. HTML and CSS: Design and Build Websites" by Jon Duckett,2011
2. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins,2018
3. HTML5 Pocket Reference" by Jennifer Niederst Robbins,2013
4. Head First HTML and CSS" by Elisabeth Robson and Eric Freeman,2012
5. CSS:The Missing Manual by David Sawyer McFarland,2015
6. HTML & CSS: The Good Parts by Ben Henick,2010.
7. HTML & XHTML :The Definitive Guide by Chuck Musciano & Bill kennedy ,2006

BCSP 236 (Basics of HTML)

Course Objectives: students will be able to:

1. Develop proficiency in creating well-structured and semantic HTML documents using appropriate tags and elements.
2. Gain hands-on experience in applying CSS styles to enhance the appearance and layout of web pages.
3. Practice implementing interactive elements such as hyperlinks, images, forms, and multimedia content in HTML.
4. Enhance understanding of CSS fundamentals including selectors, properties, and box model for effective web design.
5. Acquire skills in designing responsive and visually appealing web layouts using HTML and CSS techniques.

Credits=2	**SEMESTER-I** **BCSP 236 (Basics of HTML)**	**No. of hours per unit/ credits (30)**
	1. Create a basic HTML document with appropriate structure (DOCTYPE, html, head, body). 2. Use HTML tags to create headings (h1-h6) and paragraphs. 3. Add line breaks and horizontal rules to separate content on webpage. 4. Insert images into your HTML document using the img tag. 5. Create hyperlinks (both absolute and relative paths) to link to other web pages. 6. Insert images into your HTML document using the img tag. 7. Create ordered and unordered lists with list items. 8. Design a table with rows and columns, and apply colspan and rowspan attributes. 9. Create ordered and unordered lists with list items. 10. Design a table with rows and columns, and apply colspan and rowspan attributes. 11. Hanging Create a CSS file. 12. Apply CSS styles to the change the background color. 13. Control fonts by changing font family, size, style using CSS. 14. Experiment with CSS colors, including named colors, RGB 15. Use CSS to style block level elements (div, p) and inline elements (span, a) 16. Apply border properties to elements to create borders with different styles, widths, and colors. 17. Use padding properties to add space inside elements and adjust content positioning. 18. Apply margin properties to create spacing between elements on the webpage. 19. Experiment with advanced CSS techniques like grouping selectors and applying dimensions (width, height). 20. Create a basic webpage layout using CSS for positioning, floating, and aligning elements.	

Course Outcomes: students will be able to:

1. Demonstrate proficiency in creating well-structured HTML documents with appropriate tags and elements for effective web content presentation.

2. Showcase the ability to apply CSS styles effectively to enhance the visual appeal and formatting of HTML elements, including text, images, and backgrounds.

3. Develop skills in utilizing HTML5 semantic elements and CSS techniques for improved webpage structure and layout design.

4. Gain experience in implementing interactive features such as hyperlinks, images, tables, forms, and navigation bars using HTML and CSS.

5. Graduates of the course will be capable of designing responsive web layouts, understanding the CSS box model, and overcoming common challenges in web design and development.

Reference Books:

1. HTML and CSS: Design and Build Websites" by Jon Duckett,2011
2. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins,2018
3. HTML5 Pocket Reference" by Jennifer Niederst Robbins,2013
4. Head First HTML and CSS" by Elisabeth Robson and Eric Freeman,2012
5. CSS: The Missing Manual by David Sawyer McFarland,2015
6. HTML & CSS: The Good Parts by Ben Henick,2010.
7. HTML & XHTML: The Definitive Guide by Chuck Musciano & Bill kennedy ,2006

BCSTVSC 1: Object Oriented Programming Using C+

+ Course Objectives: Students will be able to

1. Mastering object-oriented programming principles.
2. Enhancing software development skills.
3. Leveraging C++'s efficiency for system-level programming.
4. Understanding memory management concepts.
5. Preparing for high-performance computing applications.

Credits=2	**SEMESTER-I** **Paper I: Object Oriented Programming Using C++**	**No. of hours per unit/ credits**
UNIT I	**Unit – I: Foundations of Object-Oriented Programming in C++**	**(8)**
	1.1 Concepts of OOP- C++: Terminology-Tokens, Keywords, Identifiers, constants, Operators 1.2 Basic data types 1.3 Structure of C++ program, Input and output streams 1.4 Control structures-Branching and looping statements. 1.5 Classes and objects, Access modifiers 1.6 static data members and static member function 1.7 Array of objects, passing parameter to function 1.8 This pointer, Constructors, Destructors and Inheritance 1.9 Constructors: Definition, types- Default constructor, Copy constructor, Parameterized constructor. 1.10 Destructors.	
UNIT II	**UNIT II - Advanced C++ Techniques**	**(7)**
	2.1Polymorphism-Definition, Types of polymorphism- Function Overloading 2.2 Operator Overloading-Definition, overloading unary and binary operators 2.3Overloading operators using friend function 2.4 Rules for overloading operator 2.5Virtual function 2.6Method overriding 2.7 Abstraction, Encapsulation 2.8 Pointers, Arrays 2.9 Introduction to file handling**.**	

Course Outcomes:

At the end of this course, students should be able to:

1. Proficiency in developing high-performance applications.
2. Mastery of object-oriented programming principles.
3. Ability to work on system-level programming tasks.
4. Enhancement of problem-solving skills, and preparation for careers in software development.

Reference Books:

1. "C++ Primer" by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo,2013
2. "Programming: Principles and Practice Using C++" by Bjarne Stroustrup,2008
3. "Object-Oriented Programming in C++" by Robert Lafore,2014
4. "Effective C++: 55 Specific Ways to Improve Your Programs and Designs" by Scott Meyers,2005
5. "The C++ Programming Language" by Bjarne Stroustrup,2013
6. "C++ Primer Plus" by Stephen Prata, 2019.
7. "Modern C++ Programming with Test-Driven Development: Code Better, Sleep Better" by Jeff Langr, 2013.

Lab Course on BCSTVSC 1: Object Oriented Programming using C++

Course Objectives: students will be able to:

1. Reinforce Theory with Hands-on Learning.
2. Develop Strong Coding Skills.
3. Learn Debugging and Error Handling.
4. Deepen Understanding of Algorithms.
5. Prepare for Real Software Development.

Credits=1	**SEMESTER-I** **Lab Course I (Object Oriented Programming using C++)**	**No. of hours (30)**
	1. Write a C++ program to demonstrate the use of various data types and identifiers. 2. Create a C++ program to perform basic arithmetic operations. 3. Implement a C++ program to find the factorial of a given number. 4. Write a program to calculate the area of a circle. 5. Write a C++ program to print the first 10 natural numbers using a for loop. 6. Develop a program to calculate the factorial of a number using a while loop. 7. Implement a C++ program to display the Fibonacci series up to a given limit using a do-while loop. 8. Create a program to find the sum of all even numbers between 1 and 100 using a for loop. 9. Write a C++ program to print the multiplication table of a given number using a for loop. 10. Develop a program to check whether a given number is a prime number or not using a for loop. 11. Implement a program to display the pattern. 12. Implement a C++ program to demonstrate class and objects. 13. Write a C++ program for swapping two numbers. 14. Implement operator overloading in C++. 15. . Write a C++ program to demonstrate the use of constructors (default, parameterized, copy) and destructors in a class. 16. Write a program to demonstrate the concept of inheritance with a base class "Shape" and derived classes (Rectangle, Circle) to calculate area and perimeter. 17. Develop a class "Employee" with static data members (employee count, total salary) and static member functions to display employee details and average salary. 18. Write a C++ program to implement abstraction. 19. Write a C++ program to implement Polymorphism. 20. Implement a C++ program to read and write data to a file using file handling techniques.	

Course Outcomes: 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.

Reference Books:

1. "C++ Programming: From Problem Analysis to Program Design" by D.S. Malik , 2014.
2. "Accelerated C++: Practical Programming by Example" by Andrew Koenig and Barbara E. Moo,2000.
3. "C++ How to Program" by Paul Deitel and Harvey Deitel, 2017.
4. "C++ Primer Plus" by Stephen Prata, 2019.
5. "Thinking in C++" by Bruce Eckel, 2000.
6. "Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14" by Scott Meyers 2014.
7. "The C++ Standard Library: A Tutorial and Reference" by Nicolai M. Josuttis, 2019.

BCSTSEC 2: Lab Course: System Analysis Design

Course objectives: Student will be able to

1. Broaden their knowledge of software engineering.
2. Learn Software testing algorithms and programs.
3. Provide the knowledge of Dot Net Frameworks along with C#
4. Analyse object-oriented paradigm in the C # programming language

Credits (Total Credit=2)	**Lab Course: System Analysis Design**	**No. of hours per unit/credits**
	System Analysis Design	**30**
	Exercise No.1. Preparing Software Requirements Specifications. **Exercise No.2.** Identifying Domain Classes from the Problem Statements. **Exercise No.3**. Modeling UML Class Diagrams and Sequence diagrams **Exercise No.4.** Modeling UML Use Case Diagrams and Capturing Use Case Scenarios. **Exercise No.5.** E-R Modeling **Exercise No.6.** State chart and Activity Modeling. **Exercise No.7.** Modeling Data Flow Diagrams **Exercise No.8**. Estimation of Project Metrics **Exercise No.9.** Estimation of Test Coverage Metrics and Structural Complexity **Exercise No.10**. Designing Test Suites **Exercise No.11.** Conduct unit tests for the implemented system modules. **Exercise No.12.** Perform integration testing for the developed system. **Exercise No.13**. Design an interface for integrating two disparate systems. **Exercise No.14** Evaluate the performance of a deployed system through user feedback. **Exercise No.15** Propose and implement an update to an existing system. **Exercise No.16.** Document the steps for migrating data from one system to another. **Exercise No.17.** Develop a maintenance plan for a deployed information system.	

	Exercise No.18. Troubleshoot and resolve a technical issue in a system environment. **Exercise No.19.** Present a case study of a successful system implementation. **Exercise No.20.** Participate in a group project to analyze, design, and present a solution for a real-world system problem.	

Course outcomes-Students should be able

1. Learn how to design and program Python applications.
2. Implement database and GUI applications.
3. Use the features of Dot Net Framework along with the features of C#.
4. Develop correct, well-documented programs using the C# programming language.

Practical References

1. Roger S.Pressman, Software engineering- A practitioner’s Approach, McGraw-Hill 1996
2. Pankaj Jalote- An Integrated Approach to Software Engineering, Springer Verlag, 1997.
3. Professional C# – Wrox Publication by Simon Robinson, Christain Nagel, Karli Watson, Jay Glynn, Morgan Skinner, Bill Evjen, 2001.
4. Inside C#, by Tom Archer ISBN: 0735612889 Microsoft Press Â© 2001, 403 pages Beginning ASP.NET 3.5, Wrox Publication November 2007.
5. The unified modeling language user guide Grady Booch, James Rambaugh, 1998

BCSTVEC 1: Computer Science for Environmental Awareness 2 credits

Course Objective: Student should be able to

1. Understand the environmental impact of computing technologies and their life cycle assessments.
2. Develop skills in sustainable software engineering and green coding practices to minimize resource consumption.
3. Analyze global environmental policies and ethical considerations relevant to the tech industry.
4. Apply knowledge to assess and mitigate the environmental footprint of digital products and data centers.
5. Explore opportunities for innovation and leadership in environmental sustainability within the field of computer science.

Credit=2	**SEMESTER-I** **Paper I: Environmental Awareness for Computer Science**	**No. of hours per unit / credits**
UNIT I	**Unit – I: Introduction to Environmental Awareness**	**8**
	1.1 Definition of environmental awareness in the context of computer science 1.2 Importance of integrating environmental considerations into technology development 1.3 Historical background: evolution of environmental consciousness in the tech industry 1.4 Overview of key environmental challenges (e.g., climate change, e-waste, energy consumption).	
UNIT II	**UNIT II - Environmental Impact of Computing**	**7**
	2.1Understanding the environmental footprint of computing devices (e.g., laptops, smartphones, servers) 2.2 Life cycle assessment (LCA) of digital products: extraction, production, use, and disposal phases 2.3 Carbon footprint analysis in data centres and cloud computing-waste management: challenges and solutions for proper disposal and recycling.	
UNIT III	**UNIT III: Sustainable Software Development**	**7**
	3.1Principles of sustainable software engineering: efficiency, scalability, and resource optimization 3.2 green coding practices: writing energy-efficient code, minimizing computational resources, Tools 3.3 methodologies for measuring and reducing software-related environmental impact 3.4Case studies of companies implementing sustainable software practices.	
UNIT IV	**UNIT IV: Environmental Policies and Ethical Considerations**	**8**
	4.1Overview of global environmental policies and regulations relevant to the tech industry 4.2Ethical considerations in technology development: balancing innovation with environmental responsibility 4.3 Corporate social responsibility (CSR) in tech: case studies of companies leading in environmental sustainability 4.4 Future trends and opportunities in green computing and environmental awareness.	

Course Outcomes: Student will be able to…

1Comprehend the environmental ramifications of computing devices and technologies, which involve assessing their life cycles and analyzing their carbon footprints.

2. Apply principles of sustainable software development to create energy-efficient and resource-optimized software solutions.
3. Analyze and evaluate environmental policies and regulations relevant to the tech industry, considering ethical considerations and corporate social responsibility.
4. Develop practical skills in measuring, analyzing, and reducing the environmental footprint of digital products and systems.
5. Collaborate effectively in addressing real-world environmental challenges within the context of computer science, proposing innovative solutions and promoting environmental awareness in the tech sector.

Reference Book:

1. "Greening through IT: Information Technology for Environmental Sustainability" by Bill Tomlinson - 2010.
2. "Sustainable Software Development: An Agile Perspective" by Kevin Jay Moody - 2012.
3. "Digital Disciplines: Attaining Market Leadership via the Cloud, Big Data, Mobility, and Social Media" by Joe Weinman - 2015.
4. "Environmental Politics and Policy in Industrialized Countries" by Andrew Jordan, David Vogel - .
5. "Green IT: Technologies and Applications" edited by Bhuvan Unhelkar - 2010.
6. "The Greening of IT: How Companies Can Make a Difference for the Environment" by John Lamb - 2009.
7. "Sustainable IT Architecture: The Progressive Way of Overhauling Information Systems with Minimal Impact on the Environment" by Pallab Saha -2013.

Major Syllabus

Course Structure for B.Sc. II (Semester- IV)

Theory				Practical				
Course Title	**Course Code**	**Lecture per week**	**Credits**	**Course**	**Course Title**	**Course Code**	**Lectur e per week**	**Credits**
Object Oriented Concept using Java	BCST-241	5	2	Major Practical-I	Based on BCST 241	BCSP 243	4	2
Computational Statistics I	BCST-242		2	Major Practial -II	Based on BCST 242	BCSP 244		

Note: B: B. Sc. T=Theory and P= Practical

*** B. Sc. II Semester IV***

Course Number	Course Code	Course Name
VII	BCST 241	Object Oriented Concept using Java
VIII	BCST 242	Computational Statistics I
Lab-II	BCSP 243	Practical based on BCST 241
	BCSP 244	Practical based on BCST 242

Major Syllabus

B. Sc II Semester IV

Theory: Course-I BCST 241: Object Oriented Concepts Using JAVA

Course Objectives: 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. Imbibe User Interface using Swing and AWT

Credits =2	**SEMESTER-IV** **BCST 241** **Object Oriented Concepts Using JAVA**	**No. of hours per unit/credits**
UNIT – I	**Introduction To Java, Objects and Classes**	**9**
	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	
UNIT – II	**Inheritance, Polymorphism and Packages**	**7**
	2.1 Defining sub class, subclass constructor, Inheritance-Multiple and hierarchical. 2.2 Defining packages, system packages, Creating & accessingpackages, Adding a class to package. 2.3 Polymorphism- function overloading and over ridding 2.4 Difference between method overloading and	

	overriding.	
UNIT – III	**Multithreading and Exception Handling**	**8**
	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 code, Multiple catch statement, Using finally statement, Throwing exception, user defined exception.	
UNIT – IV	**AWT and Event Handling**	**6**
	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	

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 Books:

1. Complete reference Java by Herbert Schildt, McGraw Hill, 5th edition, 2002.
2. Java 2 programming black books, Steven Horlzner, Paraglyph, Incorporated,2nd edition, 2001
3. Programming with Java, A primer, Forthedition, By E.Balagurusamy, McGraw Hill,4th Edition2010.
4. Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Sun Microsystems Press, Prentice Hall; 8th edition ,2007.
5. Java Programming- Rajendra Salokhe, Aruta Publication, First Edition, 2008.
6. Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson,2014.

Syllabus – Major II
B. Sc. II – Semester – IV
Theory: BCST 242: Computational Statistics I

Course Objectives: Students will be able to

1) Understand the technique of data collection & its presentation.
2) Learn various measures of central tendencies, dispersion, moments, skewness, kurtosis and to interpret them.
3) Describe concept of correction coefficient and how to interpret it's value. Evaluate correlation coefficients.
4) Apply concept of simple linear regression
5) Study basic concept of probability, axiomatic theory of probability, univariate probability distribution. Evaluate difference between random and non-random experiments Evaluate probabilities of different events. Apply the concept of conditional distribution.

Credits=4	**SEMESTER – IV** **BCST 242: Computational Statistics I**	**No. of hours per unit/ credits**
Credit –I UNIT I	**Data Condescension, Presentation, and Measures of Central Tendency**	**7**
	1.1 Definition and scope of Statistics, concept of statistical population sample, qualitative & quantitative data, variables. Scales of measurements: Nominal, Ordinal, Interval &Ratio. Collection and Summarization of univariate data and frequency distribution 1.2 Data Presentation: Diagrammatic & graphical presentation with real applications- Pie diagram, line diagram. Simple, multiple & partial bar diagram, histogram, ogive curves 1.3 Mathematical and positional averages: Data Presentation: M, G.M, H.M, relation between them and their properties. Median, mode ,partition values	
Credit –I UNIT II	**Measures of Dispersion and Moments**	**7**
	2.1 Measures of Dispersion: Range, Quartile deviation, Mean deviation, standard deviation, coefficient of variation. Various properties of these measures and their utility. 2.2 Raw and central moments, factorial moments, central moments in terms of raw moment's up to 4th order. Definition, Measures of skewness: Bowley's coefficient, Karl Pearson's coefficient, measure of skewness based on moment 2.3 Kurtosis: Definition, measures of kurtosis, Sheppard's correction.	
Credit –I UNIT III	**Correlation And Regression**	**8**
	3.1 Bivariate Data, Covariance, correlation, Types of correlation. Scatter diagram, Karl Pearson's coefficient of correlation (r): Definition, properties, examples, 3.2 Spearman's Rank Correlation Coefficient, (tie and without tie), 3.3 Concept of dependent and independent variables. Concept of regression, Lines of regression	

	3.4 Properties of regression coefficient	
Credit –I **UNIT IV**	**Probability**	**8**
	4.1 Concepts of experiments and random experiments. 4.2 Definitions: Sample space, Discrete sample space (finite and countably infinite), Types of Event, Power set |P(Ω. Apriori (classical) definition of probability of an event. Axiomatic definition of probability with reference to a finite and countably infinite, 4.3 Definition of conditional probability of an event. Multiplication theorem for two events. Partition of sample space. Idea of Posteriori probability, Statement and proof of Baye's theorem, examples on Baye's theorem. Elementary examples on probability and conditional probability . Concept of Independence of two events	

Course Outcomes:

At the end of this course, students should be able to:

1) learn Mathematical averages, positional averages, measure of dispersion, moments skewness and kurtosis, Types of correlation, fitting of line of Regression, sample space, power set definition of probability
2) Create Constructions of Diagrams and Graphs, Mathematical Averages and Positional Averages, Absolute and Relative measures of dispersion, Moments Skewness and Kurtosis, Bivariate data, Correlation, Regression, Random experiment, events and types of events, Conditional Probability and Independence of events.
3) Evaluate Relation between AM ,GM, HM, Derivation of Median and Mode, Properties of Measures of central tendency and dispersion, First four raw and central moments, measures of Skewness and Kurtosis, Interpretation of r if r=1, r= -1, r= 0, Properties of correlation coefficient, Derivation of the formula for Spearman's rank correlation coefficient
4) Apply fitting of regression plan by method of least square Examples on sample space, simple examples on probability based on permutation and combination, Theorems on probability.

Reference Books:

1. B. L. Agarwal, Basic Statistics (New Age International (P) Ltd., 2015) for Unit-I, II, III, IV; Unit-I: P. No. 13-41, Unit-II: P. No.42-97, Unit-III: P. No. 368-384.
2. A. M. Goon, M. K. Gupta and B. Dasgupta, Fundamentals of Statistics Vol. I and II (Calcutta: World Press, 2016) Unit-I: P. No- 42-89, Unit-II, III: P. No. 90-158.
3. S. P. Gupta, Statistical Methods (New Delhi: Sultan Chand and Sons, 2002), Unit-I: P. No. 39-61, 127-176 Unit-II: P. No. 177-335, Unit-III: P. No.337-387, Unit-IV: P. No. 495-535.
4. D. N. Elhance, Fundamentals of Statistics, (Kitab Mahal, 1978), Unit-II: P. No. 87- 177, Unit-III: P. No. 236-249.
5. Parimal Mukhopadyay, An Introduction to the Theory of Probability (World Scientific Publishing Company, 2011), Unit – I, II,: P.No. 7-84.

BCSP 243: Lab Course (Object Oriented Concepts Using JAVA)

Course Objectives: Students should be able to…

1. Improve the analytical skills of object-oriented programming and formal introduction toJava programming language.
2. Understand Object Oriented Programming language.
3. Study abnormal termination of a program using exception handling.
4. Imbibe User Interface using Swing and AWT

Credits=2	**Lab Course (Object Oriented Concepts using JAVA)**	**No. of hours per unit / credits**
	Object Oriented Concepts using JAVA	**(30)**
	Exercise No.1. Program to define a structure of a basic JAVA program. **Exercise No.2.** Program to define the data types. **Exercise No.3.** Program to define the variable. **Exercise No.4.** Program to define the operators. **Exercise No.5.** Program to define the arrays. **Exercise No.6.** Program to define the Decision-Making statements. **Exercise No.7.** Program to define the Loop statements. **Exercise No.8**. Program to define the Jump statements. **Exercise No.9.** Program to define the Type casting. **Exercise No.10**. Program to define class and constructors. Demonstrate constructors. **Exercise No.11.** Program to define class and inheritance. Demonstrate simple inheritance. **Exercise No.12.** Program to define class and inheritance. Demonstrate multilevel inheritance. **Exercise No.13**. Program to define class and inheritance. Demonstrate hierarchical inheritance.	

	Exercise No.14. Program to define class, methods, and objects. Demonstrate method overloading. **Exercise No.15**. Program to define inheritance and show method overriding. **Exercise No.16.** Program to demonstrate user define Packages. **Exercise No.17.** Program to demonstrate API Packages. **Exercise No.18.** Program to demonstrate Exception Handling. **Exercise No.19.** Program to demonstrate Multithreading. **Exercise No.20.** Program to demonstrate Applet structure and event handling.	

Course outcomes-Students should be able to

1. Show competence in the use of the Java programming language in the development.
2. of small to medium-sized application programs that demonstrate professionally
3. acceptable coding and performance standard
4. Learn the basic principles of the object-oriented programming.
5. Demonstrate an introductory understanding of graphical user interfaces, multithreaded programming, and event-driven programming.
6. Analyse professionally acceptable coding and performance standard.
7. Demonstrate graphical user interfaces, multithreaded programming, and event-drivenprogramming.

Reference Books-

1. Complete reference Java by Herbert Schildt, McGraw Hill, 5th edition, 2002.
2. Java 2 programming black books, Steven Horlzner, Paraglyph, Incorporated,2nd edition,2001
3. Programming with Java, A primer, Forthedition, By E.Balagurusamy, McGraw Hill,4thEdition2010.
4. Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell,Sun Microsystems Press, Prentice Hall; 8th edition ,2007.
5. Java Programming- Rajendra Salokhe, Aruta Publication, First Edition, 2008.
6. Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson,2014.

BCSP 244: Lab course I (Computational Statistics I)

OBJECTIVES: Student should able to

1. Understand statistical data.
2. Evaluate various measures of central tendency, dispersion, moments, Skewness and kurtosis.
3. Learn linear relationship between the two variables.
4. Predict value for dependent variable with the help of independent variable

Credits=2	**SEMESTER – IV** **Major Practical II (BCSP 244: Computational Statistics I**	**No. of hours per unit/ credits (30**
	1. Construction of Discrete Frequency Distribution. 2. Construction of continuous Frequency Distribution 3. Diagrammatic Representation. 4. Graphical Representation I 5. Graphical Representation II 6. Measure of Central Tendency-AM, GM, HM (Individual data) 7. Measure of Central Tendency- AM, GM, HM (discrete data) 8. Measure of Central Tendency- AM, GM, HM (Continuous data) 9. Measure of Central Tendency-Mode, Median Quartile (Individual data) 10. Measure of Central Tendency- Mode, Median Quartile (discrete data) 11. Measure of Central Tendency- Mode, Median Quartile (Continuous data) 12. Measure of Dispersion (Individual data) 13. Measure of Dispersion (discrete data) 14. Measure of Dispersion (Continuous data) 15. Correlation coefficient 16. Spearman’s Rank Correlation (Without tie) 17. Spearman’s Rank Correlation (With tie) 18. Regression-II 19. Application of Probability- I 20. Application of Probability - II	

Learning Outcomes: Students are able to

1) Create diagram and graphs based on frequency distribution
2) learn how to summarized data and find averages as well as spread of the data from central value (average).
3) Evaluate the coefficient of correlation between two and more variables.
4) Predicts value of one variable when other is known by using technique of regression analysis.
5) Learn the probabilities of events and conditional probabilities.

Reference Books:

1. S. P. Gupta, Statistical Methods (New Delhi: Sultan Chand and Sons, 2002), UnitI,II,III, IV: P. No. 751-803.
2. B. L. Agarwal, Basic Statistics (New Age International (P) Ltd., 2015), Unit-I : 98121.
3. S. Saxena, J. N. Kapoor, Mathematical Statistics (S Chand & Company, 2010), UnitI: P. No. 126-140, Unit-II: P. No. 179-190
4. V. K. Kapoor, Gupat S. C., Fundamental of Mathematical Statistics (S Chand & Company, 2008) , Unit-III, IV : 4.1 – 5.72
5. Parimal Mukhopadyay, An Introduction to the Theory of Probability (World Scientific Publishing Company, 2011), 183-213.

Minor Syllabus

Course Structure for B.Sc. II (Semester- IV)

Theory				Practical				
Course Title	**Course Code**	**Lecture per week**	**Credits**	**Course**	**Course Title**	**Course Code**	**Lecture per week**	**Credits**
Operating System	BCST 245	4	2	Practical -1	Based on BCS 245	BCSP -246	2	2

Note: B: B. Sc. T=Theory and P= Practical

Structure and Title of Courses of B. Sc. Course:

*** B. Sc. II Semester I V***

Course Number	Course Code	Course Name
I	BCST 245	Operating System
Lab-I	BCSP 246	Based on BCS 245

Syllabus
B. Sc II Semester IV

Course MINOR – I BCST 245: Operating Systems Course objectives: Student will be able to

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

Credits (Total Credits 2)	**SEMESTER-IV** **BCST 245** **Operating Systems**	**No. of hours per unit/ credits**
UNIT I	**Fundamental Concepts**	**(8)**
	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.	
UNIT II	**Operating System Organization**	**(7)**
	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.	
UNIT III	**Process Management and Memory Management**	**(8)**
	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 pre-emptive strategies. 3.3 Mapping address space to memory space, memory allocation strategies, fixed partition, variable partition, paging, virtual memory.	
UNIT IV	**Shell introduction and Shell Scripting**	**(7)**
	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).	

Course outcomes:

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. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, John Wiley Publications, 8th Edition, 2008.
2. A.S. Tanenbaum, Modern Operating Systems, Pearson Education, 3rd Edition, 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, Pearson Education,3rd Edition, 2003.
4. W. Stallings, Operating Systems, Internals & Design Principles, Prentice Hall of India, 5th Edition, 2008.
5. M. Milinkovic, Operating Systems- Concepts and design, Tata McGraw Hill , New Delhi, 2nd Edition, 2009.
6. System Programming and Operating System – D. M. Dhamdhere, Tata McGraw Hill, 2nd Revised Edition, 2002.
7. Unix shell programming- Yashwant Kanetkar, BPB Publications,1st edition, 2003.
8. UNIX Unbounded: A Beginning Approach by Amir Afzal, Pearson Education,3rd Edition ,2011

BCSP 246: Lab Course II Operating

Systems Course objectives: Student will be able to

1. Understand Shell operating system and memory management techniques.
2. Identify the need to create the special purpose operating system
3. Learn case studies to demonstrate practical applications of different concepts.
4. Imbibe scope to students where they can solve small, real life problems.

Credits (Total Credit=2)	**Lab Course: Operating Systems**	**No. of hours per unit/credits**
	Operating Systems	**30**
	Exercise No.1. Write a program to check status of keyboard using interrupt handler **Exercise No.2.** Write a program to implement copy command of DOS **Exercise No.3**. Write a program to display date and time of system **Exercise No.4.** Write a program to implement pwd command of linux. **Exercise No.5.** Write a program to implement wc command of linux. **Exercise No.6.** Usage of following commands: ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd. **Exercise No.7.** Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date. **Exercise No.8**. Usage of following commands: chmod, grep, tput (clear, highlight), bc. **Exercise No.9.** Write a shell script to check if the number entered at the command line is prime or not. **Exercise No.10**. Write a shell script to modify “cal” command to display calendars of the specified months **Exercise No.11.** Write a shell script to modify “cal” command to display calendars of the specified range of months.	

	Exercise No.12. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid” **Exercise No.13**. Write a shell script to display date in the mm/dd/yy format. **Exercise No.14** Write a shell script to display on the screen sorted output of “who” command along with the total number of users. **Exercise No.15** Write a shell script to display the multiplication table any number. **Exercise No.16.** Program to implement pwd command of Linux (C) **Exercise No.17.** Program to Rename a File (mv) **Exercise No.18.** Program to Check if a File Exists (test) **Exercise No.19.** Case study on case study on Shell and Various Types of Shell, Various Editors Present in Linux. **Exercise No.20.** Case study on Different Modes of Operation in Vi Editor, Shell Script, Writing and Executing Shell Scripts, Shell Variables	

Course Outcomes: Students will be able to…

1. Derive the process & memory management in Operating System and the basicOperating System organization.
2. Design programs using a Shell Scripting.
3. Analyze professionally acceptable coding and performance standard.
4. Demonstrate graphical user interfaces, multithreaded programming, and event-driven programming.

Practical Reference

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, John Wiley Publications, 8th Edition,2008.
2. A.S. Tanenbaum, Modern Operating Systems, Pearson Education ,3rd Edition, 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, Pearson Education,3rd Edition, 2003.
4. W. Stallings, Operating Systems, Internals & Design Principles, Prentice Hall of India,5th Edition, 2008.
5. M. Milinkovic, Operating Systems- Concepts and design, Tata McGraw Hill , New Delhi, 2nd Edition, 2009.
6. System Programming and Operating System – D. M. Dhamdhere,Tata McGraw Hill, 2nd Revised Edition, 2002.
7. Unix shell programming- Yashwant Kanetkar, BPB Publications, 1st edition, 2003.

Course VSC: BCSTVSC 2: Fundamental of Cyber Security

Course objectives: Student will be able to

1. Understand core cyber security principles and concepts.
2. Acquire practical knowledge of securing networks, systems, and data.
3. Identify and mitigate common cyber threats effectively.
4. Apply risk management and compliance practices in cyber security operations

Credits (Total Credit=2)	**Lab Course: Fundamental of Cyber Security**	**No. of hours per unit/credits**
	Fundamental of Cyber Security	
	Exercise No.1. Study of the features of firewall in providing network security and to set Firewall Security in windows. **Exercise No.2.** Steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome) 7. **Exercise No.3**. Study of different types of vulnerabilities for hacking a websites / Web Applications. **Exercise No.4.** Analysis the Security Vulnerabilities of E-commerce services. **Exercise No.5.** Case Study On basic technology in cyber security. **Exercise No.6.** Analysis the security vulnerabilities of E-Mail Application. **Exercise No.7.** Perform encryption and decryption of Caesar cipher Write a script for performing these operations. **Exercise No.8**. Perform encryption and decryption of a Rail fence Write a script for performing these operations. **Exercise No.9.** Case Study on – Cyber Harassment **Exercise No.10** Case Study on – Cyber Law **Exercise No.11.** Case Study on – Patent Law. **Exercise No.12.** Case Study on – Confidentiality, integrity, and availability (CIA triad) **Exercise No.13**.Study of set Firewall Security in windows.. **Exercise No.14** Basics of network security	

	Exercise No.15 Why is it important to keep software up to date?. **Exercise No.16.** Case study of Phishing Awareness: **Exercise No.17.** How do you create a strong password? **Exercise No.18.** What is social engineering, and how does it exploit human psychology? **Exercise No.19.** Case study on Data Backups: **Exercise No.20.** Case study on Define malware and describe common types of malware.	

Course Outcomes: Students will be able to…

1. Derive the process & memory management in Operating System and the basicOperating System organization.
2. Design programs using a Shell Scripting.
3. Analyze professionally acceptable coding and performance standard.
4. Demonstrate graphical user interfaces, multithreaded programming, and event-driven programming.

Practical Reference

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, John Wiley Publications , 8th Edition,2008.
2. A.S. Tanenbaum, Modern Operating Systems, Pearson Education ,3rd Edition, 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, Pearson Education,3rd Edition, 2003.
4. W. Stallings, Operating Systems, Internals & Design Principles, Prentice Hall of India,5th Edition, 2008.
5. M. Milinkovic, Operating Systems- Concepts and design, Tata McGraw Hill , New Delhi, 2nd Edition, 2009.
6. System Programming and Operating System – D. M. Dhamdhere,Tata McGraw Hill, 2nd Revised Edition, 2002.
7. Unix shell programming- Yashwant Kanetkar, BPB Publications, 1st edition, 2003.

B. Sc II Semester IV

SEC

Skill Enhancement Course (BCSTSEC 3)

XML Programming

Course Objectives: 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. Understand equipment functions and library resources related to the study of photoshop.

Credits (Total Credits 2)	**SEMESTER-IV** **Skill Enhancement Course (SEC):** **XML Programming**	**No. of hours per unit/credits (15)**
UNIT-I	**Introduction to XML**	**8**
	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,	
UNIT-II	**XML Style Sheets**	**7**
	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	

Course Outcomes: Students will be able to…

1. Learn the structure and advantages of Extensible Markup Language (XML) for data representation and exchange.
2. Explore the basics of creating XML documents, transforming XML documents, and validating XML documents.
3. Learn how to transform XML documents into documents of other types using XSLT
4. Apply the fundamentals of XQuery, its applications, and how to perform complex queries on XML data.

Reference Books:

1. "XML in a Nutshell" by Elliotte Rusty Harold and W. Scott Means (Third Edition, 2004)
2. "Beginning XML" by David Hunter, Jeff Rafter, Joe Fawcett, Eric van der Vlist, and Danny Ayers (Fourth Edition, 2007)
3. "Professional XML" by Bill Evjen, Kent Sharkey, Thiru Thangarathinam, Michael Kay, Alessandro Vernet, Sam Ferguson, and Jeremy W. Murphy (Third Edition, 2007)
4. "XML Bible" by Elliotte Rusty Harold (Third Edition, 2004)
5. "XML Pocket Reference" by Robert Eckstein (Third Edition, 2005)

Lab Course: Skill Enhancement Course (BCSTSEC 3): XML Programming

Course Objectives: 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

Credits (1)	List of Practical's	No. of Hours (30)
1	Creating a simple XML document.	
2	Creating a simple XML document with attributes	
3	Create an XML file using the Internal DTD.	
4	Create an external DTD and implement it in XML file.	
5	Creating a XML Schema.	
6	Create an xml file to implement CSS concept.	
7	To create a simple XSLT transformation from XSL to XML.	
8	To demonstrate XSL formatting-how to format numbers using the element.	
9	To demonstrate XSL formatting-how to format using element.	
10	Create a XML file to implement XPATH concept.	
11	Creating a XML DOM class.	
12	To identify the error in the XML file using the XML parser error functions.	
13	Create a well-formed XML document containing details of a car like: id, company name, model, engine and mileage.	
14	Create a valid XML document containing details of a car like: id, company name, model, engine and mileage using DTD.	
15	Create a valid XML document containing details of a car like: id, company name, model, engine and mileage using XML Schema.	
16	Create a XML document which contains details of cars and display the same as a table using XSLT.	
17	Write a Java program to parse the XML document containing car details using SAX API.	
18	Design an XML structure to represent the people in a company department. The department should have a name and an arbitrary number of staff members. Each staff member should have a name and employee ID.	
19	Creating a Wishlist for books with XSLT	
20	Write a program to connect a XML web page to any database engine.	

Course Outcomes: 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.

Reference Books:

1. Routledge (6 October 2008), Art and Design in Photoshop, 1st edition
2. Bittu Kumar (29 January 2013), Adobe Photoshop, V&S Publishers; Latest Revised Edition
3. Kogent Learning Solutions Inc. (1 January 2012), Photoshop CS6 in Simple Steps, Dreamtech Press
4. "Learning XML" by Erik T. Ray (Second Edition, 2003).

Semester – IV
Co-Curricular Course
BCSTCC 2 :Fundamental of Physical Education

Course Objectives: Students will be able to

1. Understand the components of holistic health and wellness.

2. Explore the interrelationship between physical, mental, emotional and social well-being.

3. Develop strategies for enhancing personal health and well-being.

4. Evaluate the impact of lifestyle choices on overall health.

5. Apply evidence-based practices for promoting health and preventing disease.

Credits (Total Credit=1)	**SEMESTER-I** **BCSTCC 2: Fundamental of Physical Education**	**No. of hours per unit/credits**
UNIT I	**Introduction to Physical Education**	**8**
	1.1 Definition of physical education. 1.2 History of physical education. 1.3 Principles of fitness and health and wellness 1.4 Importance of Holistic Health	
UNIT II	**Importance of Physical Education**	**8**
	2.1 Physical fitness 2.2 Stress management 2.3 Mental and physical Health 2.4 Development of physical health	
UNIT III	P**hysical Fitness and Nutrition**	**7**
	3.1 Introduction to physical Fitness and Nutrition. 3.2 Nutritional Analysis and Diet Planning Software in Computer science. 3.3 Mobile Apps for Fitness and Nutrition. 3.4 Mind-body practices (yoga, meditation, etc.)	
UNIT III	**Internet of Things (IOT) in Health and Wellness.**	**7**
	4.1IOT device for home fitness equipment. 4.2 Machine learning for Personalized Health Recommendations. 4.3 Stress reduction techniques. 4.4 Strategies for promoting mental well-being	

Course Outcomes: At the end of this course, students should be able to:
1. Learn the importance of maintaining physical health and nutrition for health and wellness.0
2. Learn techniques for managing stress and promoting mental well-being using Internet of things for health and wellness.
3. Gain knowledge about nutrition and healthy eating habits for healthy life.
4. Acquire skills in time management and getting healthy life for students.

Reference Book:

1. Shanti K Y (1987) “The Science of Yogic Breathier” (Pranayama) D B Bombay
2. Ziegler E F (2007) “An Introduction to sports and physical Education” PHilosophy Delhi
3. Pinto John and Ramachandra K (2021) Kannada Version “Dahika Sikshanda Parichaya” Louis Publications, Manglore.
4. Uppal A K & Gautam G P (2008) health and Physical Education. Friends Publication New Delhi
5. Dixit Suresh (2006) Swasthya Shiksha sports Publications Delhi.
6. Thomas D Fahey and others. Fit and Well: 6th Edition New York: McGraw Hill Publishers, 2005.
7. Puri. K. Chandra S. S. (2005) “Health and Physical Education” New Delhi: Surjeet Publication.
8. Bucher. C.A.(1979) foundation of physical Education (5th edition Missouri CV Mosby Co.
9. AAPHERD “Health related Physical Fitness Test Mannual” 1980 Published by Association drive Reston Virginia.
10. Pinto John and Roshan Kumar (2021) “Introduction to Physical Education”, Louis Publication, Mangalore.