

Bharatiya Vidya Bhavan's

**M. M. College of Arts, N.M. Institute of Science,
H.R.J. College of Commerce. (Bhavan's College)
Autonomous**

(Affiliated to University of Mumbai)

Syllabus for: S.Y.B.Sc.Computer Science

Program: B.Sc.

Program Code: BH. BSc

Course Code: (BH. USCS)

Choice Based Credit System (CBCS)

with effect from academic year 2022-23

PREAMBLE

The goal of the B.Sc. program in Computer Science is to provide students the foundations of various career opportunities in the field of Information and Communication Technology (ICT). This programme emphasizes acquisition of knowledge and understanding of the system, various programming languages and tools required for effective computation based problem solving.

The main objective of this program is to inculcate among the students, the technical as well as the theoretical knowledge about the computers and its various applications in different fields. This program is designed in such a way that students can have a detailed knowledge of subjects as well as the knowledge of information technology related applications. Throughout this program the students will go through the Information and Communication Technology (ICT) scenario, its scope, career and the essentials of the ICT world.

The Bachelor of Computer Science programme will equip the students with the necessary knowledge and skills for the existing and emerging challenges that a career in computing and software technology will entail. In addition it prepares graduates to show high quality of independent thought, flexibility and maturity based on a sound technical knowledge of the field.

BHAVAN'S COLLEGE, ANDHERI WEST

S.Y.B.Sc. COMPUTER SCIENCE Credit Based System and Grading System SYLLABUS FOR AUTONOMY

Semester – III (THEORY)				
Course Code	Course Type	Course Title	Credits	Lectures /Week
BH.USCS301	Core Subject	Theory of Computation	2	3
BH.USCS302	Core Subject	Introduction to Java Programming	2	3
BH.USCS303	Core Subject	Operating Systems	2	3
BH.USCS304	Core Subject	Data Structures	2	3
BH.USCS305	Core Subject	Combinatorics and Graph Theory	2	3
BH.USCS306	Core Subject	Software Engineering	2	3
BH.USCS307	Ability Enhancement Course 1	.NET Technologies	2	3 Hands-On
SEMESTER - III (PRACTICAL)				
Course Code	Course Type	Course Title	Credits	Lectures /Week
BH.USCSP301	Core Subject Practical	Practical of USCS301 (Web Programming Practical's)	1	3
BH.USCSP302	Core Subject Practical	Practical of USCS302	1	3
BH.USCSP303	Core Subject Practical	Practical of USCS303	1	3
BH.USCSP304	Core Subject Practical	Practical of USCS304	1	3
BH.USCSP305	Core Subject Practical	Practical of USCS305	1	3
BH.USCSP306	Core Subject Practical	Practical of USCS306	1	3

Semester – IV (THEORY)				
Course Code	Course Type	Course Title	Credits	Lectures /Week
BH.USCS401	Core Subject	Advanced Database Management	2	3
BH.USCS402	Core Subject	Advanced Java Programming	2	3
BH.USCS403	Core Subject	Computer Networks	2	3
BH.USCS404	Core Subject	Physical Computing and IoT Programming	2	3
BH.USCS405	Core Subject	Linear Algebra using Python	2	3
BH.USCS406	Core Subject	Cyber Forensics	2	3
BH.USCS407	Ability Enhancement Course 2	Android Developer Fundamentals	2	3 Hands-On
BH.USCSP401	Core Subject Practical	Practical of USCS401	1	3
BH.USCSP402	Core Subject Practical	Practical of USCS402	1	3
BH.USCSP403	Core Subject Practical	Practical of USCS403	1	3
BH.USCSP404	Core Subject Practical	Practical of USCS404	1	3
BH.USCSP405	Core Subject Practical	Practical of USCS405	1	3
BH.USCSP406	Core Subject Practical	Practical of USCS406	1	3

PROGRAM OUTCOMES

The B.Sc. programme is formulated based on the inputs received from the members of Board of Studies of Bhavan's College (Autonomous) and is geared meets the standards prescribed by general science education. Our students are allowed to choose from any of the three subjects from the cluster of physics, chemistry botany, zoology, statistics mathematics and microbiology. Some learning outcomes include:

PO1.	Disciplinary Knowledge: Capable of demonstrating comprehensive knowledge and understanding of one or more other disciplines that form a part of an undergraduate programme of study. This programme helps students in building a solid foundation for further higher studies and research.
PO2.	Critical Thinking & Competency in Skills: Take informed actions after identifying the assumptions that frame our thinking and actions. Critically evaluate practices, policies and theories by following scientific approach to knowledge development. Obtain proficiency in analytical reasoning, critical understanding, analysis and synthesis in order to solve theoretical and practical problems. This can orient students towards applications of their subject in other disciplines and, can also be utilized in modelling and solving real life problems.
PO3.	Communication Skills: Ability to express thoughts and ideas effectively in writing and orally; communicate with others using appropriate media; confidently share one's views and express herself/ himself; demonstrate the ability to listen carefully; and present complex information in a clear and concise manner to different groups.
PO4.	Social Interaction: Ability to work effectively and respectfully with diverse teams; facilitate cooperative or coordinated effort on the part of a group and act together as a group or a team in the interests of a common cause. Elicit views of others, mediate disagreements and help reach conclusions in group settings.
PO5.	Environment and Sustainability: Understand the issues of environmental contexts and sustainable development.
PO6.	Interdisciplinary and Research Skills: A sense of inquiry and capability for asking relevant/ appropriate questions, Ability to recognize cause- and- effect relationships, define problems, formulate hypotheses, interpret and draw conclusions from data, ability to plan, execute and report the results of an experiment or investigation which will enable them to apply one's learning to real life situations.
PO7.	Proficiency in Employments: This programme will help students to enhance their employability for Government jobs, related to science, data analysis jobs, and jobs in various other public and private enterprises.

PROGRAM SPECIFIC OUTCOME

PSO	Description
	A student completing Bachelor’s Degree in Computer Science program will be able to:
PSO-1	<p>Foundation of mathematical concepts : obtain the Ability to apply mathematical methodologies to solve computation tasks, model real world problems using appropriate data structure and suitable algorithms.</p> <p>Foundations of Software development : obtain the Ability to understand the structure and development methodologies of software systems. Possess professional skills of software design process.</p>
PSO-2	<p>Applications of Computing and Research Ability: obtain the Ability to use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovation</p> <p>Problem analysis: Identify, formulate, review research literature, and analyze complex technical problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and computer sciences</p>
PSO-3	<p>Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.</p> <p>Project management and finance: Demonstrate knowledge and understanding of the development and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments</p>
PSO-4	<p>Design/development of solutions: Design solutions for complex technical problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.</p>
PSO-5	<p>Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information</p>

	<p>to provide valid conclusions.</p> <p>Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the software development practice</p>
PSO-6	<p>Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern technical methods and IT tools including prediction and modeling with an understanding of the limitations.</p> <p>Communication: Communicate effectively on complex technical activities with the development community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.</p>

DETAILED SYLLABUS

SEMESTER III

Theory of Computation

Programme: B.Sc.(CS)			Semester: III		
Course: Theory of Computation			Course Code: BH.USCS301		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<u>Pre-requisites:</u> 1. Basic knowledge of computing.					
<u>Course Objectives: The learner should be able -</u> A. To provide comprehensive insight into theory of computation by understanding grammar, languages and other elements of modern language design. B. To develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas. C. To develop capability to design and construct Turing Machines.					
<u>Course Outcomes: After successfully completing the course, the learner-</u> A. Understands Automata theory, Grammar and its application in Language Design B. Learns about context-free languages and its application with Pushdown Automata. C. Understands Linear Bound Automata and Turing Machines					
INDEX					
Unit	Description			Periods	
1	Automata Theory and Formal Languages			15 L	
2	Regular Sets and Regular Grammar, Context Free Languages, Pushdown Automata			15 L	
3	Linear Bound Automata, Turing Machines, Undecidability			15 L	

	Total	45 L
--	--------------	-------------

Detailed Syllabus		
Unit	Detailed Description	Lecture/Period
I	<p>Automata Theory: Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata.</p> <p>Formal Languages: Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata</p>	15 Lectures
II	<p>Regular Sets and Regular Grammar: Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar</p> <p>Context Free Languages: Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG</p> <p>Pushdown Automata: Definitions, Acceptance by PDA, PDA and CFG</p>	15 Lectures
III	<p>Linear Bound Automata: The Linear Bound Automata Model, Linear Bound Automata and Languages.</p> <p>Turing Machines: Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine</p> <p>Undecidability: The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvability Problems</p>	15 Lectures
	Total	45 Lectures
<p><u>Text Book:</u></p> <ol style="list-style-type: none"> 1. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI, 3rd Edition 2. Introduction to Computer Theory, Daniel Cohen, Wiley, 2nd Edition 3. Introductory Theory of Computer Science, E.V. Krishnamurthy, Affiliated East-West Press 		
<p><u>Reference Books:</u></p> <ol style="list-style-type: none"> 1. Theory of Computation, Kavi Mahesh, Wiley India 2. Elements of The Theory of Computation, Lewis, Papadimitriou, PHI 3. Introduction to Languages and the Theory of Computation, John E Martin, McGraw- 		

Hill Education

4. Introduction to Theory of Computation, Michel Sipser, Thomson

Self-Study topics:

1. Differentiate between NFA and DFA.
2. Implementation of Automata theory in computing.
3. Acceptability by Turing Machines.
4. Explore more Halting Problems.

List of Topics for the practicals for Web Programming:

1. Design a webpage that makes use of
 - a. Document Structure Tags
 - b. Various Text Formatting Tags
 - c. List Tags
 - d. Image and Image Maps
2. Design a webpage that makes use of
 - a. Table tags
 - b. Form Tags (forms with various form elements)
 - c. Navigation across multiple pages
 - d. Embedded Multimedia elements
3. Design a webpage that make use of Cascading Style Sheets with
 - a. CSS properties to change the background of a Page
 - b. CSS properties to change Fonts and Text Styles
 - c. CSS properties for positioning an element
4. Write JavaScript code for
 - a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
 - b. Validating the various Form Elements
5. Write JavaScript code for
 - a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date
 - b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document,
 - c. Storing and Retrieving Cookies
6. Create a XML file with Internal / External DTD and display it using
 - a. CSS
 - b. XSL
7. Design a webpage to handle asynchronous requests using AJAX on
 - a. Mouseover
 - b. button click
8. Write PHP scripts for
 - a. Retrieving data from HTML forms
 - b. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
 - c. Working with Arrays
 - d. Working with Files (Reading / Writing)
9. Write PHP scripts for
 - a. Working with Databases (Storing Records / Repriving Records and Display

<p>them)</p> <p>b. Storing and Retrieving Cookies</p> <p>c. Storing and Retrieving Sessions</p> <p>10. Design a webpage with some jQuery animation effects</p>
<p><u>Details of Conduct of Practical Examination (Evaluation Scheme):</u></p> <p>40 -Marks practical work</p> <p>05 -Marks Viva</p> <p>05 -Marks Journal</p>

Introduction to Java Programming

Programme: B.Sc.(CS)			Semester: III		
Course: Introduction to Java Programming			Course Code: BH.USCS302		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<p><u>Pre-requisites:</u></p> <ol style="list-style-type: none"> 1. Basic concepts of object oriented programming. 2. Basic programming constructs. 3. Requirement of object oriented programming. 					
<p><u>Course Objectives:</u></p> <ol style="list-style-type: none"> A. To teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and to cover-up with the prerequisites of Core java. B. To introduce Java's Graphical application development. C. To understand Java's concepts of packages, interfaces, exception handling, file handling, multithreading etc. 					
<p><u>Course Outcomes: After successfully completing the course</u></p> <ol style="list-style-type: none"> A. Object oriented programming concepts using Java. Also knowledge of input, its processing and getting suitable output. B. Understand, design, implement and evaluate classes and GUI programming using swing package. C. Knowledge and implementation packages and Interface, Serialization, multithreading, file 					

handling, exception handling etc.

INDEX

Unit	Description	Periods
1	The Java Language, OOP, String manipulation, Packages and interfaces, Inheritance	15 L
2	Exception handling, File handling, Multithreading, Networking	15 L
3	Swing, JDBC	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	<p>The Java Language: Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays.</p> <p>OOPS: Introduction, Class, Object, Static Keywords, Constructors, this Keyword, Inheritance, super Keyword, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces.</p> <p>String Manipulations: String, String Buffer, String Tokenizer.</p> <p>Packages: Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers.</p>	15 Lectures
II	<p>Exception Handling: Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples</p> <p>Multithreading: Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods.</p> <p>I/O Streams: Introduction, Byte-oriented streams, Character-</p>	15 Lectures

	oriented streams, File, Random access File, Serialization. Networking: Introduction, Socket, Server socket, Client –Server Communication	
III	Swing: Need for swing components, Components hierarchy, Panes, Swing components: JLabel, JTextField and JPasswordField, JTextArea, JButton, JCheckBox, JRadioButton, JComboBox and JList JDBC: Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updates, CallableStatement, BLOB & CLOB	15 Lectures
	Total	45 Lectures
<u>Text Book:</u>		
<ol style="list-style-type: none"> Herbert Schildt, Java The Complete Reference, Ninth Edition, McGraw-Hill Education, 2014 Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR, 9th Edition. 		
<u>Reference Books:</u>		
<ol style="list-style-type: none"> E. Balagurusamy, Programming with Java, Tata McGraw-Hill Education India, 2014 Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press The Java Tutorials: http://docs.oracle.com/javase/tutorial/ Advanced Java Programming, Uttam K. Roy, Oxford University Press The Java Tutorials of Sun Microsystems Inc. 		
<u>Self-Study topics:</u>		
<ul style="list-style-type: none"> Java execution environment. JVM working. How does Java differ from other languages? Java's collection framework. Java wrapper classes. Java Inner classes. Importance of java's swing programming. 		
<u>List of Topics for the practicals:</u>		
<ol style="list-style-type: none"> Accept integer values for a, b and c which are coefficients of quadratic equations. Find the solution of the quadratic equation. Accept two n x m matrices. Write a Java program to find addition of these matrices. Accept n strings. Sort names in ascending order. Create a package: Animals. In the package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package animals. Demonstrate Java inheritance using extends keyword. Demonstrate method overloading and method overriding in Java. 		

<ol style="list-style-type: none"> 7. Demonstrate creating your own exception in Java. 8. Java programming on file handling. 9. Java programming on multithreading. 10. Java socket programming. 11. Using various swing components, I design a Java application to accept a student's resume. (Design form) 12. Develop a Java application using swing components. 13. Develop a Java application to store and retrieve data from a database. 14. Develop a Java application to store images in a database as well as retrieve images from the database.
<p><u>Details of Conduct of Practical Examination (Evaluation Scheme):</u></p> <p>40 -Marks practical work 05 -Marks Viva 05 -Marks Journal</p>

Operating Systems

Programme: B.Sc.(CS)			Semester: III		
Course: Operating Systems			Course Code: BH.USCS303		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<p><u>Pre-requisites:</u></p> <ol style="list-style-type: none"> 1. Basic understanding of the working of the computer 2. Knowledge of programming in any high level programming language 					
<p><u>Course Objectives:</u></p> <ol style="list-style-type: none"> A. To understand proper working of operating systems. B. To provide a sound understanding of the Computer operating system, its structures and functioning. C. To understand what a process is and how processes are synchronized and scheduled; and how memory is managed 					

Course Outcomes: After successfully completing the course, the learner will have:

- A. Understanding the proper working of operating systems.
- B. Understanding of the Computer operating system, its structures and functioning.
- C. Learner will have understanding of processes and how they are synchronized; and understanding of memory management

INDEX

Unit	Description	Periods
1	Introduction and Operating-Systems Structures, Operating-System Structures, Processes, Threads	15 L
2	Process Synchronization, CPU Scheduling, Deadlocks	15 L
3	Main Memory, Virtual Memory, Mass-Storage Structure, File-System Interface, File-System Implementation	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	Introduction and Operating-Systems Structures: Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments Operating-System Structures: Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure Processes: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication Threads: Overview, Multicore Programming, Multithreading Models	15 Lectures
II	Process Synchronization: General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling	15 Lectures

	Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	
III	Main Memory: Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing Mass-Storage Structure: Overview, Disk Structure, Disk Scheduling, Disk Management File-System Interface: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing File-System Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management	15 Lectures
	Total	45 Lectures

Text Book:

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 8th Edition

Reference Books:

1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill
2. Naresh Chauhan, Principles of Operating Systems, Oxford Press
3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016

Self-Study topics:

1. Operating System's role
2. Computing Environments
3. File System Interface

List of Topics for the practicals:

(Practical can be implemented either in Java or any other programming language.)

1. Implement FCFS scheduling algorithm in Java.
2. Implement SJF (with no preemption) scheduling algorithm in Java
3. Implement RR scheduling algorithm in Java
4. Write a Java program that implements the banker's algorithm
5. Write a Java program that implements the FIFO page-replacement algorithm
6. Write a Java program that implements the LRU page-replacement algorithm.
7. Design a File System in Java.

8. Process Communication:

- (i) Implement the producer-consumer problem using shared memory.
- (ii) Implement the producer-consumer problem using message passing.
- (iii) One form of communication in a Client-Server Systems environment is Remote

method invocation (RMI). RMI is a Java feature similar to RPCs. RMI allows a thread to invoke a method on a remote object. Objects are considered remote if they reside in a different Java virtual machine (JVM). Demonstrate RMI program for adding/subtracting/multiplying/dividing two numbers.

9. Threads:

(i) The Java version of a multithreaded program that determines the summation of a non-negative integer. The Summation class implements the Runnable interface. Thread creation is performed by creating an object instance of the Thread class and passing the constructor a Runnable object.

(ii) Write a multithreaded Java program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.

10. Synchronization:

(i) Implement Bounded buffer problem in Java.

(ii) Implement the readers-writers problem using Java synchronization.

(iii) The Sleeping-Barber Problem: A barber shop consists of a waiting room with n chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers using Java synchronization.

Details of Conduct of Practical Examination (Evaluation Scheme):

40 -Marks practical work

05 -Marks Viva

05 -Marks Journal

Data Structures

Programme: B.Sc.(CS)			Semester: III		
Course: Data Structures			Course Code: BH.USCS304		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<u>Pre-requisites:</u>					

1. Procedural and Object Oriented Programming in C++
2. Knowledge of the basic working of a computer.

Course Objectives:

- A. To explore and understand the concepts of Data Structures and its significance in programming.
- B. To provide a holistic approach to design, use and implement abstract data types.
- C. To understand basic principles of algorithm design and why algorithm analysis is important.

Course Outcomes: After successfully completing the course, the learner will have

- A. Knowledge about Data structures, its types and significance in programming
- B. Deeper understanding of Abstract Data types and their implementation
- C. Understanding of the concepts of algorithms for designing a good program

INDEX

Unit	Description	Periods
1	Algorithm Analysis, Arrays, Linked Lists	15 L
2	Stacks, Queues	15 L
3	Hash Tables, Trees	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	<p><u>Algorithm Analysis:</u> Variables, Data types, data structures, ADT, Algorithms, Analysis of Algorithms, Running time analysis, Asymptotic Notation (O, Θ, Ω)</p> <p><u>Array applications:</u> Searching - Linear Search, Binary Search, Sorting - Bubble, Selection, Insertion, Merge, Quick. Analysis of all sorting techniques</p> <p><u>Linked Lists:</u> Self referential structures, List as a Data Structure, differences with array. Dynamic implementation of Linked List, Types of Linked List – Singly(all operations), Doubly(create, display), Circular Singly(create, display) and circular doubly(create, display); Time complexity of operations. Applications of Linked List – polynomial representation, Addition</p>	15 Lectures

	of two polynomials	
II	<p>Stacks: Introduction, Representation- Static & Dynamic, Primitive Operations on stack, Application of Stack, Evaluation of postfix and prefix expressions, Conversion of expressions(infix to prefix and infix to postfix)</p> <p>Queues: Introduction, Representation - Static & Dynamic, Primitive Operations on Queue, Application of Queue, Types of queues - Circular queue, priority queue, deque</p>	15 Lectures
III	<p>Trees: Introduction, definition, Glossary, Types of trees, Operations on binary and binary search tree, Tree Traversals(inorder, pre-order, postorder), Balanced Binary Search Trees, Height balanced tree- AVL trees- Rotations, AVL tree examples.</p> <p>Graphs: Concept & terminologies, Graph Representation – Adjacency matrix, list. Degree of Graph(indegree, outdegree), Traversals – BFS and DFS, Applications – AOV network – topological sort, AOE network – critical Path</p>	15 Lectures
	Total	45 Lectures

Text Book:

1. Fundamentals of Data Structures - Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran
2. Computer Algorithms - Ellis Horowitz, Sartaj Sahni

Reference Books:

1. Data Structures using C and C++ By Yedidyah Langsam, Aaron M. Tenenbaum, Moshe J. Augenstein
2. Introduction to Data Structures using C By Ashok Kamthane
3. Data Structures using C Bandopadhyay & Dey (Pearson)
4. Data Structures using C By Srivastava BPB Publication.

Self-Study topics:

1. Finding the time complexity of all algorithms studied
2. Analysis of all sorting techniques
3. AVL tree examples, Applications of AVL trees
4. Applications – AOV network – topological sort, AOE network – critical path

List of Topics for the practicals:

- 1) Implement Linear Search to find an item in a list.
- 2) Implement Binary Search to find an item in an ordered list.
- 3) Implement Sorting Algorithms
 - a. Bubble sort
 - b. Insertion sort

- c. Quick sort
- d. Merge Sort
- e. Selection sort
- f. Radix Sort
- 4) Implement Linked list and demonstrate the functionality to add, delete, search for items in the linked list.(singly, doubly, doubly circular list)
- 5) Implement working of Stacks(all primitive operations).
- 6) Implement Program for
 - a. Infix to Postfix conversion
 - b. Postfix Evaluation
- 7) Implement the following
 - a. Static implementation of circular queue of integers with following operations:- Initialize(),insert(), delete(), isempty(), isfull(), display()
 - b. Dynamic implementation of Queue for integers with the following operations:
 - Insert
 - Delete
 - Display
 - Exit
 - c. Dynamic implementation of circular queue of integers with following operations:
 - Initialize(),insert(), delete(), isempty(), isfull(), display()
- 8) Implement Binary Search Tree and its traversals.
- 9) Implement AVL Trees
- 10) Implement BFS and DFS in a graph

Details of Conduct of Practical Examination (Evaluation Scheme):

40 -Marks practical work
 05 -Marks Viva
 05 -Marks Journal

Combinatorics and Graph Theory

Programme: B.Sc.(CS)			Semester: III		
Course: Combinatorics and Graph Theory			Course Code: BH.USCS305		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)

3	3	1	2+1	(Marks: 40)	(Marks: 60)
Pre-requisites: Basic of computing.					
Course Objectives: The learner should be able - A. To get a broad exposure of combinatorial Mathematics B. To have understanding of combinatorics and Graph theory and its applications. C. To gain knowledge of combinatorial theorems and applications.					
Course Outcomes: After successfully completing the course, the learner - A. Appreciates the beauty of combinatorics and how combinatorial problems naturally arise in many settings. B. Understands the combinatorial and Graph theory features in real world situations and Computer Science applications. C. Applies combinatorial and graph theoretical concepts to understand Computer Science concepts and apply them to solve problems					
INDEX					
Unit	Description			Periods	
1	Introduction to Combinatorics			15 L	
2	Graph Theory			15 L	
3	Network Flows			15 L	
	Total			45 L	

Detailed Syllabus		
Unit	Detailed Description	Lecture/Period
I	Introduction to Combinatorics: Enumeration, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, Sudoku Puzzles. Strings, Sets, and Binomial Coefficients: Strings- A First Look, Combinations, Combinatorial, The Ubiquitous Nature of Binomial Coefficients, The Binomial, Multinomial Coefficients. Induction: Introduction, The Positive Integers are Well Ordered, The Meaning of Statements, Binomial Coefficients Revisited, Solving Combinatorial Problems Recursively, Mathematical Induction, and Inductive Definitions Proofs by Induction. Strong	15 Lectures

	Induction	
II	Graph Theory: Basic Notation and Terminology, Multigraphs: Loops and Multiple Edges, Eulerian and Hamiltonian Graphs, Graph Coloring, Planar Counting, Labeled Trees, A Digression into Complexity Theory. Applying Probability to Combinatorics, Small Ramsey Numbers, Estimating Ramsey Numbers, Applying Probability to Ramsey Theory, Ramsey's Theorem The Probabilistic Method	15 Lectures
III	Network Flows: Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm, A Concrete Example, Integer Solutions of Linear Programming Problems. Combinatorial Applications of Network Flows: Introduction, Matching in Bipartite Graphs, Chain partitioning, Pólya's Enumeration Theorem: Coloring the Vertices of a Square.	15 Lectures
	Total	45 Lectures

Text Book:

1. Applied Combinatorics, Mitchel T. Keller and William T. Trotter, 2016, <http://www.rellek.net/appcomb>.

Reference Books:

1. Applied Combinatorics, sixth edition, Alan Tucker, Wiley; (2016)
2. Graph Theory and Combinatorics, Ralph P. Grimaldi, Pearson Education; Fifth edition (2012)
3. Combinatorics and Graph Theory, John Harris, Jeffry L. Hirst, Springer(2010).
4. Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008).

Self-Study topics:

As per instructor's guidance.

List of Topics for the practicals:

1. Solving problems on strings, sets and binomial coefficients.
2. Solving problems using induction.
3. Solving problems on Eulerian and Hamiltonian graphs.
4. Solving problems on Chromatic number and coloring
5. Solving problems using Kruskal's Algorithm
6. Solving problems using Prim's Algorithm
7. Solving problems using Dijkstra's Algorithm
8. Solving problems of finding augmenting paths in network flows.
9. Solving problems on network flows using Ford-Fulkerson Labeling Algorithm
10. Solving problems on posets and their associated networks.

Details of Conduct of Practical Examination (Evaluation Scheme):

40 -Marks practical work
05 -Marks Viva
05 -Marks Journal

Software Engineering

Programme: B.Sc.(CS)			Semester: III		
Course: Software Engineering			Course Code: BH.USCS306		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<p><u>Pre-requisites:</u></p> <ol style="list-style-type: none"> 1. Basic concepts of a software. 2. Basics of designing software. 3. Role of testing in software development. 					
<p><u>Course Objectives:</u></p> <ol style="list-style-type: none"> A. To teach the learner to gather requirements and understand the nature of it for developing models. B. To introduce the basic concepts of designing a software with various methods. C. To teach the learner, how testing methods can be used as effective tools in providing quality assurance concerning software with test plan / case design. 					
<p><u>Course Outcomes: After successfully completing the course</u></p> <ol style="list-style-type: none"> A. Learners understand the software process including requirement gathering with model development. B. Learner understands the designing of a software with various software testing, metrics methods and strategies. C. Learner understands to design test cases and identify defects and manage those for assuring the quality of a software. 					
INDEX					
Unit	Description			Periods	
1	Introduction, Requirement Analysis and System Modeling			15 L	
2	System Design, Software Measurement and Metrics, Software Project Management, Project Scheduling			15 L	

3	Risk Management, Software Quality Assurance, Software Testing	15 L
	Total	45 L

Detailed Syllabus		
Unit	Detailed Description	Lecture/Period
I	<p>Introduction: The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming</p> <p>Requirement Analysis and System Modeling: Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS , Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram</p>	15 Lectures
II	<p>System Design: System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design</p> <p>Software Measurement and Metrics: Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality</p> <p>Software Project Management: Estimation in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision</p> <p>Risk Management - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan</p>	15 Lectures
III	<p>Software Testing and Introduction to quality: Introduction, Nature of errors, an example for Testing, Definition of Quality ,</p>	15 Lectures

	<p>QA, QC, QM and SQA , Software Quality Factors</p> <p>Quality Costs and Improvement : Defining Quality Costs, Types of Quality Costs, Introduction QI, Pareto Diagrams, Cause-effect Diagrams.</p> <p>Software Quality Assurance: Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, Software Reliability, The ISO 9000 Quality Standards, Capability Maturity Model</p> <p>Software Testing : Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design</p>	
	Total	45 Lectures

Text Book:

1. Software Engineering, A Practitioner’s Approach, Roger S, Pressman.(2014)

Reference Books:

1. Software Engineering, Ian Sommerville, Pearson Education
2. Software Engineering: Principles and Practices”,Deepak Jain,OXFORD University Press,
3. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI
4. Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons
5. A Concise Introduction to Software Engineering, Pankaj Jalote, Springer

Self-Study topics:

1. Study and analyze live projects around you.
2. Software Testing Strategies : Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing.
3. Scatter Diagrams, Run charts.
4. Quality Cost Measurement, Utilizing Quality Costs for Decision-Making.

List of Topics for the practicals:

1. Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.
2. Conduct a test suite for any two web sites.
3. Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP.
4. Write and test a program to login to a specific web page.
5. Write and test a program to update 10 student records into tables into Excel files.
6. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).
7. Write and test a program to provide a total number of objects present / available on the page.
8. Write and test a program to get the number of items in a list / combo box.
9. Write and test a program to count the number of check boxes on the page checked and unchecked count.
10. Load Testing using JMeter, Android Application testing using Appium Tools, Bugzilla

Bug tracking tools
<u>Details of Conduct of Practical Examination (Evaluation Scheme):</u> 40 -Marks practical work 05 -Marks Viva 05 -Marks Journal

.NET Technology

Programme: B.Sc.(CS)		Semester: III		
Course: .NET Technologies		Course Code: BH.USCS307		
Teaching Scheme		Evaluation Scheme (Theory)		
Hands-On-Sessions (Periods per week)	Tutorial (Periods per week- per batch)	Credits (Theory +Practical)	Internal Assessment	End Semester Examination (Activity Based)
3	1	3	(Marks: 50)	(Marks: 50)
<u>Pre-requisites:</u> 1. Basics object oriented programming language, html and javascript. 2. Basic knowledge of controls. 3. Knowledge of xml.				
<u>Course Objectives:</u> A. To teach the learner how to use programming language and understand the concept .NET. B. To introduce the web controls and how to validate the web forms. C. To explore .NET technologies for designing and developing dynamic, interactive and responsive web applications and introduce Ajax.				
<u>Course Outcomes: After successfully completing the course</u> A. The learner understands the .NET framework and develop a proficiency in the C# programming language. B. The learner explores the web controls and use of validations. C. The learner understands to use ADO.NET for data persistence to develop interactive web applications and exploring Ajax.				
INDEX				
Unit	Description	Periods		

1	The .NET Framework, C# Language Basics,ASP.NET, HTML Server Controls.	15 L
2	Web Controls, State Management, Validation, Rich Controls, Themes and Master Pages, Website Navigation.	15 L
3	ADO.NET, Data Binding, Data Controls, Working with XML, Caching, LINQ, ASP.NET AJAX.	15 L
	Total	45 L

Detailed Syllabus		
Unit	Detailed Description	Lecture/Period
I	<p>The .NET Framework: .NET Languages, Common Language Runtime, .NET Class Library</p> <p>C# Language Basics: Comments, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes</p> <p>ASP.NET: Creating Websites, Anatomy of a Web Form - Page Directive, Doctype, Writing Code - Code-Behind Class, Adding Event Handlers, Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders,</p> <p>HTML Server Controls - View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File, web.config File</p>	15 Lectures
II	<p>Web Controls: Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle</p> <p>State Management: ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State</p> <p>Validation: Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions</p> <p>Rich Controls: Calendar Control, AdRotator Control, MultiView Control</p>	15 Lectures

	<p>Themes and Master Pages: How Themes Work, Applying a Simple Theme, Handling Theme Conflicts, Simple Master Page and Content Page, Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths</p> <p>Website Navigation: Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control</p>	
III	<p>ADO.NET: Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access.</p> <p>Data Binding: Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource</p> <p>Data Controls: GridView, DetailsView, FormView</p> <p>Working with XML: XML Classes – XMLTextWriter, XMLTextReader</p> <p>Caching: When to Use Caching, Output Caching, Data Caching</p> <p>LINQ: Understanding LINQ, LINQ Basics,</p> <p>ASP.NET AJAX: ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes</p>	15 Lectures
	Total	45 Lectures

Text Book:

1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)

Reference Books:

1. The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill.
2. Beginning ASP.NET 4 in C# and VB Imar Spaanjaars, WROX.

Self-Study topics:

1. Concept of c# programming language.
2. Design page using Controls.
3. Performing ADO.NET data access in ASP.NET.
4. Interacting with XML documents.

List of Topics for the hands-on practicals:

1. Create a website of your name that shows your information.
2. Implement a basic calculator with and without functions.
3. Design a web form to conduct placements for graduates in Mumbai or Pune that includes student details in textbox, course name in radio button, also including the passing year and print the if student is eligible or not based on the criteria provided.
4. Design a Login Form containing Username and password.
5. Create a form that accepts Principal Amount, No.Of Years, Rate Of Interest to calculate “Interest” and “Final Amount” on the click of the button.
6. Create a Reservation form for Mumbai- to – Pune journey containing 3 text fields to enter names, 3 text fields to enter age and a text field to show Final bill which should also contain a radio button showing the type of journey (A.C/ Non-A.C). Charges for

both are fixed but only for children with age <5 & senior citizens with age > 60, the rates are half. 4% service charges are applied on the Final Amount.

7. Perform the following activities
 - a. Design ASP.NET page and perform validation using various Validation Controls.
 - b. Design ASP.NET Pages with Rich Controls (Calendar / Ad Rotator / Multiview).
 - c. Design ASP.NET Pages with various Navigation Controls.
8. Design an APS.NET master web page and use it on other (at least 2-3) content pages.
9. Design ASP.NET Pages for State Management using
 - a. Cookies
 - b. Session State.
10. Performing ADO.NET data access in ASP.NET for data binding.
11. Design ASP.NET application for Interacting (Reading / Writing) with XML documents,
12. Design ASP.NET application to query a Database
13. Design and use AJAX based ASP.NET pages.

Modality of Assessment

Theory Examination Pattern:

A. Internal Assessment (50%) : 50 Marks

<u>S.No.</u>	<u>Evaluation Type</u>	<u>Total Marks</u>
1.	Conduct 5 Assignments (for 10 marks each) throughout the course of the syllabus. These assignments may be any of the following: Recall base or Analysis Based.	50

B. External Examination(50%) : 50 Marks

1. A project should be done using Visual Studio (using c#). Report of the same has to be prepared which should follow a specific format that includes a font size of 12, Times New Roman and where the headings should have a font size 14. This report should be hardbound/spiral bound.
2. The project can be done individually or a group of 2-3 students.
3. The students will have to present the project during the examination.
4. A certified copy of the project report is essential to appear during the examination.
5. Project will be evaluated based on the below-given criteria :
 - a. Documentation-10 Marks
 - b. Designing- 10 Marks
 - c. Working: 10 Marks
 - d. Presentation of the project- 10 Marks
 - e. Viva -10 Marks

Project Implementation Guidelines

1. A learner is expected to prepare a project in Visual Studio (using c#).
2. A learner can choose any topic which is covered in the syllabus or any other topic with the prior approval from the project in-charge.
3. The project can be performed individually or in groups of maximum 2-3.

4. The project will be a simple Web application with concepts of asp.net involved.
5. **It has to be an implemented work; theoretical study will not be acceptable.**
6. The quality of the project will be evaluated based on the novelty of the topic, scope of the work, relevance to the computer science, adoption of emerging techniques/technologies and its real-world application.
7. A learner has to maintain a project report with the following subsections:
 - a. Title Page
 - b. Certificate
The certificate should contain the following information –
 - The fact that the student has successfully completed the project as per the syllabus and that it forms a part of the requirements for completing the BSc degree in computer science of University of Mumbai.
 - The name of the student and the project guide.
 - The academic year in which the project is done.
 - Date of submission,
 - Signature of the project guide and the head of the department with date along with the department stamp,
 - c. Index Page detailing description of the following with their subsections:
 - Title: A suitable title giving the idea about what work is proposed.
 - Introduction: An introduction to the topic giving proper background of the topic.
 - Requirement Specification: Specify Software/hardware/Technologies used.
 - Results: Tables/Figures/Graphs/Screenshots/Reports etc.
 - Conclusion and Future Scope: Specify the Final conclusion and future scope
 - References: Books, web links, research articles, books referred etc.
8. The size of the project report shall be around ten to fifteen five pages, excluding the code.
9. The Project should be certified by the concerned Project guide and Head of the department.
10. A learner has to make a presentation of the working project and will be evaluated as per the Project evaluation scheme given below:
 - a. Presentation of the project : 10 Marks
 - b. Working of the project : 10 Marks
 - c. Quality of the project : 10 Marks
 - d. Viva :10 Marks
 - e. Documentation : 10 Marks

(Project will not be accepted without a Certified Project Document at the time of Project Presentation)

DETAILED SYLLABUS

SEMESTER IV

Advanced Database Management

Programme: B.Sc.(CS)			Semester: IV		
Course: Advanced Database Management			Course Code: BH.USCS401		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Contin uous Intern al Assess ment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks : 40)	(Marks: 60)
<u>Pre-requisites:</u> 1. Basic knowledge of database systems. 2. Basic query processing concepts. 3. Core concepts of database management system.					
<u>Course Objectives:</u> A. To understand and explore next generation database management technology. B. To explore BigData, MongoDB,NoSQL. C. To explore jQuery and JSON.					
<u>Course Outcomes: After successfully completing the course</u> A. Students shall be able to perform basic query processing on MongoDB. B. Students shall be able to differentiate between BigData, MongoDB and other Database systems. C. Students shall be able to work on JQuery and JSON.					
INDEX					
Unit	Description			Periods	
1	Big Data,NoSQL,Introducing MongoDB, Using			15 L	

	MongoDB Shell	
2	MongoDB Architecture, MongoDB Storage Engine, MongoDB Use Cases	15 L
3	jQuery, JSON	15 L
	Total	45 L

Detailed Syllabus		
Unit	Detailed Description	Lecture/Period
I	<p>Big Data: Getting Started, Big Data, Facts About Big Data, Big Data Sources, Three Vs of Big Data, Volume, Variety, Velocity, Usage of Big Data, Visibility, Discover and Analyze Information, Segmentation and Customizations, Aiding Decision Making, Innovation, Big Data Challenges, Policies and Procedures, Access to Data, Technology and Techniques, Legacy Systems and Big Data, Structure of Big Data, Data Storage, Data Processing, Big Data Technologies.</p> <p>NoSQL: SQL, NoSQL, Definition, A Brief History of NoSQL, ACID vs. BASE, CAP Theorem (Brewer's Theorem), The BASE, NoSQL Advantages and Disadvantages, Advantages of NoSQL, Disadvantages of NoSQL, SQL vs. NoSQL Databases, Categories of NoSQL Databases.</p> <p>Introducing MongoDB: History, MongoDB Design Philosophy, Speed, Scalability, and Agility, Non-Relational Approach, JSON-Based Document Store, Performance vs. Features, Running the Database Anywhere, SQL Comparison</p> <p>Using MongoDB Shell: Basic Querying, Create and Insert, Explicitly Creating Collections, Inserting Documents Using Loop, Inserting by Explicitly Specifying _id, Update, Delete, Read, Using Indexes, Stepping Beyond the Basics, Using Conditional Operators, Regular Expressions, MapReduce, aggregate(), Designing an Application's Data Model, Relational Data Modeling and Normalization, MongoDB Document Data Model Approach</p>	15 Lectures
II	<p>MongoDB Architecture: Core 12 20 Processes, mongod, mongo, mongos, MongoDB Tools, Standalone Deployment, Replication,</p>	15 Lectures

	<p>Master/Slave Replication, Replica Set, Implementing Advanced Clustering with Replica Sets, Sharding, Sharding Components, Data Distribution Process, Data Balancing Process, Operations, Implementing Sharding, Controlling Collection Distribution (Tag-Based Sharding), Points to Remember When Importing Data in a Sharded Environment, Monitoring for Sharding, Monitoring the Config Servers, Production Cluster Architecture, Scenario 1, Scenario 2, Scenario 3, Scenario 4</p> <p>MongoDB Storage Engine: Data Storage Engine, Data File (Relevant for MMAPv1), Namespace (.ns File), Data File (Relevant for WiredTiger), Reads and Writes, How Data Is Written Using Journaling, GridFS – The MongoDB File System, The Rationale of GridFS, GridFS under the Hood, Using GridFS, Indexing, Types of Indexes, Behaviors and Limitations</p> <p>MongoDB Use Cases: Use Case 1 - Performance Monitoring, Schema Design, Operations, Sharding, Managing the Data, Use Case 2 – Social Networking, Schema Design, Operations, Sharding</p>	
III	<p>jQuery: Introduction, Traversing the DOM, DOM Manipulation with jQuery, Events, Ajax with jQuery, jQuery Plug-ins, jQuery Image Slider.</p> <p>JSON: Introduction, JSON Grammar, JSON Values, JSON Tokens, Syntax, JSON vs XML, Data Types, Objects, Arrays, Creating JSON, 12 21 JSON Object, Parsing JSON, Persisting JSON, Data Interchange, JSON PHP, JSON HTML, JSONP</p>	15 Lectures
	Total	45 Lectures
<p><u>Text Book:</u></p> <ol style="list-style-type: none"> 1. Practical MongoDB Shakuntala Gupta Edward Navin Sabharwal Apress 2. Beginning jQuery Jack Franklin Russ Ferguson Apress Second 3. Next Generation Databases Guy Harrison Apress. 4. Beginning JSON Ben Smith Apress 		
<p><u>Reference Books:</u></p> <ol style="list-style-type: none"> 1. MongoDB Tutorials: https://docs.mongodb.com/manual/tutorial/ 2. JSON Tutorial: https://www.javatpoint.com/json-tutorial 3. jQuery Tutorial: https://www.javatpoint.com/jquery-tutorial 		
<p><u>Self-Study topics:</u></p> <ol style="list-style-type: none"> 1. Explore query processing on different real time databases. 2. Differentiate between various database management systems. 3. Explore java, PHP and python with MongoDB, jQuery and JSON. 		
<p><u>List of Topics for the practicals for Advanced Database Management:</u></p>		

1. MongoDB Basics queries to create and drop (database & collection), insert, update and delete a document.
2. Simple Queries with MongoDB.
3. Implementing Aggregation Queries with MongoDB.
4. Write a MongoDB query for Replication, Backup and Restore of database.
5. Java and MongoDB: Connecting Java with MongoDB and inserting, retrieving, updating and deleting.
6. PHP and MongoDB: Connecting PHP with MongoDB and inserting, retrieving, updating and deleting.
7. Python and MongoDB: Connecting Python with MongoDB and inserting, retrieving, updating and deleting.
8. Programs on Basic jQuery, jQuery Events, jQuery Selectors, jQuery Hide and Show effects, jQuery fading effects, jQuery Sliding effects.
9. Programs on Advanced jQuery:
 - a. jQuery Animation effects, jQuery Chaining
 - b. jQuery Callback, jQuery Get and Set Contents
 - c. jQuery Insert Content, jQuery Remove Elements and Attribute
10. JSON: Creating JSON, Parsing JSON, Persisting JSON.
11. Create a JSON file and import it to MongoDB
 - a. Export MongoDB to JSON.
 - b. Write a MongoDB query to delete JSON object from MongoDB

Details of Conduct of Practical Examination (Evaluation Scheme):

40 -Marks practical work
 05 -Marks Viva
 05 -Marks Journal

Advanced Java Programming

Programme: B.Sc.(CS)			Semester: IV		
Course: Advanced Java Programming			Course Code: BH.USCS402		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture	Practical (Periods)	Tutorial	Credits	Continuous	End Semester

(Periods per week)	per week per batch)	(Periods per week per batch)	(Theory +Practical)	Internal Assessment (CIA)	Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)

Pre-requisites:

1. To know basic core java concepts.
2. OOP concepts.
3. Java swing programming, Networking etc.

Course Objectives:

- A. To explore advanced topics of Java programming for solving problems.
- B. To explore and understand use of Java Server Programming.
- C. To implement Struts, JSON, Hibernate and JPA.

Course Outcomes: After successfully completing the course

- A. Understand the concepts related to Java Technology.
- B. Perform Java Server Programming
- C. Working with servlets,beans etc.

INDEX

Unit	Description	Periods
1	Understanding Java EE, Java EE Architecture, Server and Containers, Introduction To Java ServerPages,Getting Started With Java ServerPages, Action Elements:	15 L
2	Java Beans,Struts 2,JSON	15 L
3	Persistence, Object/Relational Mapping And JPA,Introduction to JavaPersistence API,Introduction to Hibernate,Writing Hibernate Application	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	Understanding Java EE: What is an Enterprise Application?	15 Lectures

	<p>What Is Java Enterprise Edition? Java EE Technologies, JavaEE evolution, Glassfish Server.</p> <p>Java EE Architecture, Server and Containers: Types of System Architecture, Java EE Server, Java EE Containers.</p> <p>Introduction to Java Servlets: TheNeedforDynamicContent, JavaServletTechnology, WhyServlets? WhatcanServletsdo? Servlet API and Lifecycle: JavaServletAPI, TheServletSkeleton, TheServletLifeCycle, ASimpleWelcomeServlet</p> <p>WorkingwithServlets: GettingStarted, UsingAnnotationsInsteadofDeploymentDescriptor</p> <p>Introduction To Java ServerPages: WhyuseJava ServerPages? DisadvantagesOfJSP, JSPv\Servlets, LifeCycleofaJSPPage, HowdoesaJSPfunction? HowdoesJSPexecute? AboutJava ServerPages</p> <p>Getting Started With Java ServerPages: Comments, JSPDocument, JSPElements, JSPGUIExample.</p> <p>Action Elements: IncludingotherFiles, ForwardingJSPPageToAnotherPage, PassingParametersforotherActions, Loading a Javabean.</p>	
<u>II</u>	<p>Java Beans: Introduction, JavaBeans Properties, Examples</p> <p>Struts 2: Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL</p> <p>JSON: Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java</p>	15 Lectures
<u>III</u>	<p>Persistence, Object/Relational Mapping And JPA: What Is Persistence? Persistence in Java, Current Persistence Standards In Java, Why another Persistence Standards? Object/Relational Mapping.</p> <p>Introduction to JavaPersistence API: The Java Persistence API, JPA,ORM,Database and Application, Architecture of JAVA, How JPA Works? JPA Specifications.</p> <p>Writing JPA Application: Application Requirement Specifications, Software Requirements, The Application Development Approach, Creating Database and Tables in Mysql, creating aWebApplication, Adding the Required Library Files, creating a Javabean Class, Creating Persistence Unit[Persistence.Xml], CreatingJSPS, The JPA Application Structure, Running the JPA Application.</p> <p>Introduction to Hibernate: What is Hibernate? Why Hibernate? Hibernate,Database and Application, Components of Hibernate, Architecture of Hibernate, How Hibernate Works?</p> <p>Writing Hibernate Application: Application Requirement</p>	15 Lectures

	Specifications, Software Requirements, The Application Development Approach, Creating Database and Tables in Mysql, creating Web Application, Adding the Required Library Files, creating a Javabean Class, Creating Hibernate Configuration File, Adding a Mapping Class, Creating JSPs, Running The Hibernate Application.	
	Total	45 Lectures

Text Book:

1. Java EE 7 For Beginners Sharanam Shah, Vaishali Shah.SPD, First edition 2017.
2. Advanced Java Programming Uttam Kumar Roy Oxford Press 2015
3. Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR,9th Edition.
4. Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5th Edition
5. Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3rd Edition

Reference Books:

1. Java EE 8 Cookbook: Build reliable applications with the most robust and mature technology for enterprise development Elder Moraes Packt First 2018
2. The Java Tutorials: <http://docs.oracle.com/javase/tutorial/>)
3. The Java Tutorials of Sun Microsystems Inc.

Self-Study topics:

1. Creating small hibernate based applications.
2. Develop small JPA applications.
3. Differentiate between Persistence, Object/Relational Mapping And JPA.
4. To explore more Struts 2 and JSON features.
5. Develop small scale applications based on servlets and JSP.

List of Topics for the practicals:

1. Develop the following Simple Servlet applications.
 - Write a Java application to demonstrate servlet life cycle.
 - Create a servlet for a login page.
 - Create a registration servlet in Java using JDBC.
 - Design database for student administration. Develop servlet(s) to perform CRUD operations.
2. Develop the following JSP applications.
 - Create Employees table in EMP database. Perform select, insert, update, and delete operations on the Employee table using JSP.
 - Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types.
 - Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button).
 - Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC.
 - Write a student class with three properties. The use Bean action declares a

<p>JavaBean for use in a JSP. Write a Java application to access JavaBeans Properties.</p> <ol style="list-style-type: none"> 3. Design application using Struts2. Application must accept user name and greet user when command button is pressed. 4. Write Java application to encoding and decoding JSON in Java. 5. Develop the following JPA applications. <ul style="list-style-type: none"> ● Develop a simple Inventory Application Using JPA. ● Develop a Guestbook Application Using JPA. ● Create a simple JPA application to store and retrieve Book details. 6. Develop the following JPA applications with ORM and Hibernate. <ul style="list-style-type: none"> ● Develop a JPA Application to demonstrate use of ORM associations. ● Develop a Hibernate application to store Feedback of Website Visitors in MySQL Database. ● Develop a Hibernate application to store and retrieve employee details in MySQL Database. 7. Develop the following Hibernate applications. <ul style="list-style-type: none"> ● Develop an application to demonstrate Hibernate One- To -One Mapping Using Annotation. ● Develop Hibernate application to enter and retrieve course details with ORM Mapping. ● Develop a five page web application site using any two or three Java EE Technologies.
<p><u>Details of Conduct of Practical Examination (Evaluation Scheme):</u></p> <p>40 -Marks practical work 05 -Marks Viva 05 -Marks Journal</p>

Computer Networks

Programme: B.Sc.(CS)			Semester: IV		
Course: Computer Networks			Course Code: BH.USCS403		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<u>Pre-requisites:</u>					
1. Basics of computing.					

2. Basic means of communication.
3. Role of networking in computing.

Course Objectives:

- A. Build an understanding of the fundamental concepts of data communication and computer networking.
- B. Know about routing mechanisms and different routing protocols.
- C. Know about different application layer protocols.

Course Outcomes: After successfully completing the course

- A. Learners will be able to understand the concepts of networking, which are important for them to be known as ‘networking professionals’.
- B. Learners will be able to understand and build the skills of subnetting and routing mechanisms.
- C. They will be familiar with different protocols of the application layer in networking.

INDEX

Unit	Description	Periods
1	Introduction Network Models	15 L
2	Introduction to Physical Layer and Data-Link Layer	15 L
3	Network layer, Transport Layer	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, standards and administration Internet Standards. Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture Layers in 15L the TCP/IP Protocol Suite, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing. Detailed introduction to Physical Layer, Detailed introduction to	15 Lectures

	Data-Link Layer, Detailed introduction to Network Layer, Detailed introduction to Transport Layer, Detailed introduction to Application Layer. Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay).	
II	Digital Transmission digital-to-digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM), Transmission Modes, Parallel Transmission, Serial Transmission. Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-to-analog Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM), Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time-Division Multiplexing. Transmission Media, Guided Media, Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching, Introduction to Data-Link Layer, Nodes and Links, Services, Two Sub-layers, Three Types of addresses, Address Resolution Protocol (ARP). Error Detection and Correction, introduction, Types of Errors, Redundancy, Detection versus Correction,	15 Lectures
III	Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA. Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer Switches, Routers, Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing. Unicast Routing, General Idea, Least-Cost Routing, Routing Algorithms, Distance-Vector Routing, Link-State Routing, Path-Vector Routing, Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection-Oriented Protocols. Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.	15 Lectures
	Total	45 Lectures
<u>Text Book:</u>		
<ol style="list-style-type: none"> 1. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013. 2. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson 		

Education, 2011.
<p><u>Reference Books:</u></p> <ol style="list-style-type: none"> 1. Computer Network, Bhushan Trivedi, Oxford University Press. 2. Data and Computer Communication, William Stallings, PHI.
<p><u>Self-Study topics:</u> As per instructor's guidance.</p>
<p><u>List of Topics for the practicals:</u></p> <ol style="list-style-type: none"> 1. Understanding the working of NIC cards, Ethernet/Fast Ethernet/Gigabit Ethernet. 2. Crimping of Twisted-Pair Cable with RJ45connector for Straight-Through, Cross-Over, Roll-Over. 3. To understand their respective role in networks/internet. 4. Problem solving with IPv4, which will include the concept of Classful addressing. (supportive Hint: use Cisco Binary Game) 5. Using linux-terminal or Windows-cmd, execute the following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig. 6. Using Packet Tracer, create a basic network of two computers using appropriate network wire. 7. Using Packet Tracer, connect multiple (min.6) computers using layer 2 switch. 8. Using Packet Tracer, connect a network in triangular shape with three layer two switches and every switch will have four computers. Verify their connectivity with each other. 9. Using Packet Tracer, create a wireless network of multiple PCs using appropriate access points. 10. Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working
<p><u>Details of Conduct of Practical Examination (Evaluation Scheme):</u></p> <p>40 -Marks practical work 05 -Marks Viva 05 -Marks Journal</p>

Physical Computing and Internet of Things

Programme: B.Sc.(CS)			Semester: IV		
Course: Physical Computing and Internet of Things			Course Code: BH.USCS404		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per	Practical (Periods per	Tutorial (Periods per	Credits (Theory	Continuous Internal	End Semester Examination

week)	week per batch)	week per batch)	+Practical)	Assessment (CIA)	(ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)

Pre-requisites: Basics of electronics and computing.

Course Objectives:

- A. To learn about SoC architectures.
- B. To know how Raspberry Pi works.
- C. Learn to program Raspberry Pi. Implementation of the Internet of Things and Protocols.

Course Outcomes: After successfully completing the course

- A. Enable learners to understand System On Chip Architectures.
- B. Introduction and preparing Raspberry Pi with hardware and installation.
- C. Learn physical interfaces and electronics of Raspberry Pi and program them using practical's also to learn how to make consumer grade IoT safe and secure with proper use of protocol.

INDEX

Unit	Description	Periods
1	SoC and Raspberry P	15 L
2	Programming Raspberry Pi	15 L
3	Introduction to IoT	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	<p>SoC and Raspberry Pi</p> <p>System on Chip: What is System on chip? Structure of System on Chip.</p> <p>SoC products: FPGA, GPU, APU, Compute Units.</p> <p>ARM 8 Architecture: SoC on ARM 8. ARM 8 Architecture Introduction</p>	15 Lectures

	<p>Introduction to Raspberry Pi: Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi.</p> <p>Raspberry Pi Boot: Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.</p>	
II	<p>Programming Raspberry Pi</p> <p>Raspberry Pi and Linux: About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands</p> <p>Programming interfaces: Introduction to Node.js, Python.</p> <p>Raspberry Pi Interfaces: UART, GPIO, I2C, SPI</p> <p>Useful Implementations: Cross Compilation, Pulse Width Modulation, SPI for Camera.</p>	15 Lectures
III	<p>Introduction to IoT: What is IoT? IoT examples, Simple IoT LED Program.</p> <p>IoT and Protocols IoT Security: HTTP, UPnp, CoAP, MQTT, XMPP.</p> <p>IoT Service as a Platform: Clayster, Thinger.io, SenseIoT, carriers and Node RED.</p> <p>IoT Security and Interoperability: Risks, Modes of Attacks, Tools for Security and Interoperability.</p>	15 Lectures
	Total	45 Lectures
<p><u>Text Book:</u></p> <ol style="list-style-type: none"> 1. Learning Internet of Things, Peter Waher, Packt Publishing(2015) 2. Mastering the Raspberry Pi, Warren Gay, Apress(2014) 		
<p><u>Reference Books:</u></p> <ol style="list-style-type: none"> 1. Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly. 		
<p><u>Self-Study topics:</u> As per instructor's guidance.</p>		
<p><u>List of Topics for the practicals:</u></p> <ol style="list-style-type: none"> 1. Preparing Raspberry Pi: Hardware preparation and Installation. 2. Linux Commands: Exploring the Raspbian. 3. GPIO: Light the LED with Python . 4. GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas 5. SPI: Camera Connection and capturing Images using SPI. 		

6. Real Time Clock display using PWM.
7. Stepper Motor Control: PWM to manage stepper motor speed.
8. Node RED: Connect LED to Internet of Things .
9. Stack of Raspberry Pi for better Computing and analysis.
10. Create a simple Web server using Raspberry Pi.

Details of Conduct of Practical Examination (Evaluation Scheme):

- 40 -Marks practical work
- 05 -Marks Viva
- 05 -Marks Journal

Linear Algebra using Python

Programme: B.Sc.(CS)			Semester: IV		
Course: Linear Algebra using Python			Course Code: BH.USCS405		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<u>Pre-requisites:</u>					
<ol style="list-style-type: none"> 1. Knowledge of linear algebra. 2. Basics of python programming. 					
<u>Course Objectives:</u>					
<ol style="list-style-type: none"> A. To offer the learner the relevant linear algebra concepts through computer application.. B. To explore python implementation. C. To improvise computational thinking of the learner. 					
<u>Course Outcomes: After successfully completing the course</u>					
<ol style="list-style-type: none"> A. The learner understands linear algebra in the field of computer science. B. The learner understands the concepts through programming implementations. C. The learner instills computational thinking while learning linear algebra. 					
INDEX					
Unit	Description			Periods	

1	Field	15 L
2	Matrix, Basis	15 L
3	Gaussian elimination, Inner Product, Orthogonalization, Eigenvector	15 L
	Total	45 L

Detailed Syllabus		
Unit	Detailed Description	Lecture/Period
I	<p>Field: Introduction to complex numbers, numbers in Python , Abstracting over fields, Playing with GF(2), Vector Space: Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations. Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise</p>	15 Lectures
II	<p>Matrix: Matrices as vectors, Transpose, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product, From function inverse to matrix inverse</p> <p>Basis: Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Linear dependence, Basis , Unique representation, Change of basis, first look, Computational problems involving finding a basis</p> <p>Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator</p>	15 Lectures
III	<p>Gaussian elimination: Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers,</p> <p>Inner Product: The inner product for vectors over the reals, Orthogonality,</p> <p>Orthogonalization: Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement,</p> <p>Eigenvector: Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and</p>	15 Lectures

	eigenvectors, Coordinate representation in terms of eigenvectors, The Internet worm, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.	
	Total	45 Lectures

Text Book:

1. Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)

Reference Books:

1. Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012). 2)
2. Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition (2007). 3)
3. Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition (2002)

Self-Study topics:

As per instructor's guidance.

List of Topics for the practicals:

1. Write a program which demonstrates the following:
 - Addition of two complex numbers
 - Displaying the conjugate of a complex number
 - Plotting a set of complex numbers
 - Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number $a=1/2$, $a=1/3$, $a=2$ etc.
2. Write a program to do the following:
 - Enter a vector u as a n -list
 - Enter another vector v as a n -list
 - Find the vector $au+bv$ for different values of a and b
 - Find the dot product of u and v
3. Write a program to do the following:
 - Enter two distinct faces as vectors u and v .
 - Find a new face as a linear combination of u and v i.e. $au+bv$ for a and b in \mathbb{R} .
 - Find the average face of the original faces.
4. Write a program to do the following:
 - Enter an r by c matrix M (r and c being positive integers)
 - Display M in matrix format
 - Display the rows and columns of the matrix M
 - Find the scalar multiplication of M for a given scalar.
 - Find the transpose of the matrix M .
5. Write a program to do the following:
 - Find the vector –matrix multiplication of a r by c matrix M with a c -vector u .

<ul style="list-style-type: none"> ● Find the matrix-matrix product of M with a c by p matrix N. <p>6. Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse.</p> <p>7. Write a program to convert a matrix into its row echelon form.</p> <p>8. Write a program to do the following:</p> <ul style="list-style-type: none"> ● Enter a positive number N and find numbers a and b such that $a^2 - b^2 = N$ ● Find the gcd of two numbers using Euclid's algorithm. <p>9. Write a program to do the following:</p> <ul style="list-style-type: none"> ● Enter a vector b and find the projection of b orthogonal to a given vector u. ● Find the projection of b orthogonal to a set of given vectors <p>10. Write a program to enter a given matrix and an eigenvalue of the same. Find its eigenvector.</p>
<p><u>Details of Conduct of Practical Examination (Evaluation Scheme):</u></p> <p>40 -Marks practical work 05 -Marks Viva 05 -Marks Journal</p>

Cyber Forensics

Programme: B.Sc.(CS)			Semester: IV		
Course: Cyber Forensics			Course Code: BH.USCS406		
Teaching Scheme			Evaluation Scheme (Theory)		
Lecture (Periods per week)	Practical (Periods per week per batch)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Continuous Internal Assessment (CIA)	End Semester Examination (ESE)
3	3	1	2+1	(Marks: 40)	(Marks: 60)
<p><u>Pre-requisites:</u></p> <ol style="list-style-type: none"> 1. Basics of computing and security. 2. Importance of security in the software industry. 					
<p><u>Course Objectives: The learner should be able -</u></p> <ol style="list-style-type: none"> A. To understand the procedures for identification, preservation, and extraction of data from electronic evidence for computer and network forensics. B. To have exposure regarding auditing and investigation of email systems, detecting intrusions over social networking and browser analysis. 					

C. To gain knowledge of laws, regulations and acts regarding Digital Forensics and how to analyse and verify the evidence.

Course Outcomes: After successfully completing the course, the learner

- A. Plans and prepares for all stages of an investigation.
- B. Becomes responsible while using social media platforms and alert of different types of web threats.
- C. Report cyber crime in a way that would be acceptable in the court of law and deal with cyber crime evidence in a proper manner.

INDEX

Unit	Description	Periods
1	Computer Forensics, Network Forensic, Cell Phone and Mobile Device Forensics	15 L
2	Internet Forensic, E-mail Forensics, Messenger Forensics, Social Media Forensics, Browser Forensics	15 L
3	Investigation, Evidence presentation and Legal aspects of Digital Forensics, Introduction to Legal aspects of Digital Forensics	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Period
I	<p>Computer Forensics : Introduction to Computer Forensics and standard procedure, Incident Verification and System Identification ,Recovery of Erased and damaged data, Disk Imaging and Preservation, Data Encryption and Compression, Automated Search Techniques, Forensics Software</p> <p>Network Forensic : Introduction to Network Forensics and tracking network traffic, Reviewing Network Logs, Network Forensics Tools, Performing Live Acquisitions, Order of Volatility, Standard Procedure</p> <p>Cell Phone and Mobile Device Forensics: Overview, Acquisition Procedures for Cell Phones and Mobile Devices</p>	15 Lectures

II	Internet Forensic : Introduction to Internet Forensics, World Wide Web Threats, Hacking and Illegal access, Obscene and Indecent transmission, Domain Name Ownership Investigation, Reconstructing past internet activities and events E-mail Forensics : e-mail analysis, e-mail headers and spoofing, Laws against e-mail Crime, Messenger Forensics: Yahoo Messenger Social Media Forensics: Social Media Investigations Browser Forensics: Cookie Storage and Analysis, Analyzing Cache and temporary internet files, Web browsing activity reconstruction	15 Lectures
III	Investigation, Evidence presentation and Legal aspects of Digital Forensics: Authorization to collect the evidence , Acquisition of Evidence, Authentication of the evidence, Analysis of the evidence, Reporting on the findings, Testimony Introduction to Legal aspects of Digital Forensics: Laws & regulations, Information Technology Act, Giving Evidence in court, Case Study – Cyber Crime cases, Case Study – Cyber Crime cases	15 Lectures
	Total	45 Lectures

Text Book:

1. Guide to computer forensics and investigations, Bill Nelson, Amelia Philips and Christopher Steuart, course technology, 5th Edition, 2015

Reference Books:

1. Incident Response and computer forensics, Kevin Mandia, Chris Prorise, Tata McGrawHill, 2nd Edition, 2003

Self-Study topics:

As per instructor's guidance.

List of Topics for the practicals:

1. Creating a Forensic Image using FTK Imager/Encase Imager :
 - Creating Forensic Image
 - Check Integrity of Data
 - Analyze Forensic Image
2. Data Acquisition:
 - Perform data acquisition using:
 - USB Write Blocker + Encase Imager
 - SATA Write Blocker + Encase Imager
 - Falcon Imaging Device
3. Forensics Case Study:
 - Solve the Case study (image file) provide in lab using Encase Investigator or Autopsy
4. Capturing and analyzing network packets using Wireshark (Fundamentals) :

- Identification the live network
 - Capture Packets
 - Analyze the captured packets
5. Analyze the packets provided in lab and solve the questions using Wireshark :
 - What web server software is used by www.snopes.com?
 - What cell phone problem is the client concerned about?
 - According to Zillow, what instrument will Ryan learn to play?
 - How many web servers are running Apache?
 - What hosts (IP addresses) think that jokes are more entertaining when they are explained?
 6. Using Sysinternals tools for Network Tracking and Process Monitoring :
 - Check Sysinternals tools
 - Monitor Live Processes
 - Capture RAM
 - Capture TCP/UDP packets
 - Monitor Hard Disk
 - Monitor Virtual Memory
 - Monitor Cache Memory
 7. Recovering and Inspecting deleted files
 - Check for Deleted Files
 - Recover the Deleted Files
 - Analyzing and Inspecting the recovered files

Perform this using recovery option in ENCASE and also Perform manually through command line
 8. Acquisition of Cell phones and Mobile devices
 9. Email Forensics
 - Mail Service Providers
 - Email protocols
 - Recovering emails
 - Analyzing email header
 10. Web Browser Forensics
 - Web Browser working
 - Forensics activities on browser
 - Cache / Cookies analysis
 - Last Internet activity

Details of Conduct of Practical Examination (Evaluation Scheme):

40 -Marks practical work
 05 -Marks Viva
 05 -Marks Journal

Android Developer Fundamentals

Programme: B.Sc.(CS)	Semester: IV
Course: Android Developer Fundamentals	Course Code: BH.USCS407

Teaching Scheme		Evaluation Scheme (Theory)		
Hands-On Sessions (Periods per week)	Tutorial (Periods per week per batch)	Credits (Theory +Practical)	Internal Evaluation	End Semester Examination (Activity Based)
3	1	3	(Marks: 50)	(Marks: 50)

Pre-requisites: Knowledge of Java programming language with experience in concepts of Object Oriented Programming.

Course Objectives:

- A. To provide comprehensive insight into developing applications running on smart mobile devices and demonstrate programming skills for managing tasks on mobile.
- B. To provide systematic approach for studying definition, methods and its applications for Mobile-App development
- C. To learn about basic methods, tools and techniques for developing Android Apps

Course Outcomes: After successfully completing the course

- A. The learner will be able to have comprehensive insight into developing applications running on smart mobile devices and demonstrate programming skills for managing tasks on mobile.
- B. The learner will be able to provide systematic approach for studying definition, methods and its applications for Mobile-App development
- C. The learner will be able to understand the requirements of the Mobile programming environment using Android.

INDEX

Unit	Description	Periods
1	Introduction to Android Programming	15 L
2	User Experience, Working in the Background	15 L
3	Saving user data	15 L
	Total	45 L

Detailed Syllabus

Unit	Detailed Description	Lecture/Peri
------	----------------------	--------------

		od
I	Introduction to Android Programming: What is Android? Obtaining the required tools, creating first android app, understanding the components of screen, adapting display orientation, action bar, Activities and Intents, Activity Lifecycle and Saving State, Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, ProgressBar View, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View	15 Lectures
II	User Experience: User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Material design, Providing resources for adaptive layouts, Working in the background: AsyncTask and AsyncTaskLoader, Connecting to the Internet, Broadcast receivers, Services, Notifications, Alarm managers, Transferring data efficiently	15 Lectures
III	Saving user data: Data - saving, retrieving, and loading: Overview to storing data, Shared preferences, SQLite primer, store data using SQLite database, ContentProviders, loaders to load and display data, Permissions, performance and security, Firebase and AdMob, Publish your app	15 Lectures
	Total	45 Lectures

Theory reference:

1. <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/en/android-developer-fundamentals-course-concepts-en.pdf>

Developer Guide:

1. <https://developer.android.com/guide>

Concept Lessons:

1. <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/index.html>

Hands-on training

1. <https://developer.android.com/courses/fundamentals-training/overview-v2>
2. <https://developer.android.com/courses/fundamentals-training/toc-v2>

Video reference

1. https://www.youtube.com/playlist?list=PLlyCyjh2pUe9wv-hU4my-Nen_SvXIzxGB

Self-Study topics:

1. Firebase
2. Admob
3. Publishing your app

List of Topics for the hands-on practise:

1. Install Android Studio and Run Hello World Program.
2. Create an android app with Interactive User Interface using Layouts.
3. Create an android app that demonstrates working with TextView Elements.
4. Create an android app that demonstrates Activity Lifecycle and Instance State.

5. Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers.
6. Create an android app that demonstrates the use of an Options Menu.
7. Create an android app that demonstrates Screen Navigation Using the App Bar and Tabs.
8. Create an android app to demonstrate AsyncTask and use BroadcastReceiver.
9. Create an android app to show Notifications and Alarm manager.
10. Create an android app to save user data in a database.

Modality of Assessment

Theory Examination Pattern:

A. Internal Assessment(50%) : 50 Marks

<u>S.No.</u>	<u>Evaluation Type</u>	<u>Total Marks</u>
1.	Conduct 5 Assignments (for 10 marks each) throughout the course of the syllabus. These assignments may be any of the following: Case study based, Recall based, Analysis Based	50

B. External Examination(50%) : 50 Marks

1. A project should be done based on Android Studio (using Java). A project report should be prepared. The report should have a font size of 12, Times New Roman and 1.5 line spacing. The headings should have font size 14. The report should be hardbound/spiral bound. [Project report should contain]
2. The project can be done individually or a group of two students.
3. The students will have to present the project during the examination.
4. A certified copy of the project report is essential to appear for the examination.
5. Project will be evaluated based on the below-given criteria
 - a. Documentation-10 Marks
 - b. Data Collection Methodology- 10 Marks
 - c. Results and Inferences: 10 Marks
 - d. Presentation of the project- 10 Marks
 - e. Viva voice-10 Marks

Project Implementation Guidelines

1. A learner is expected to prepare a project in Android Studio(using Java).
2. A learner can choose any topic which is covered in Semester I - semester IV or any other topic with the prior approval from the project in-charge.
3. The project can be performed individually or in groups of maximum 2.
4. The project can be application oriented/web-based/database/research based.
5. **It has to be an implemented work; just theoretical study will not be acceptable.**
6. The quality of the project will be evaluated based on the novelty of the topic, scope of the work, relevance to the computer science, adoption of emerging techniques/ technologies and its real-world application.

7. A learner has to maintain a project report with the following subsections
 - f. Title Page
 - g. Certificate

The certificate should contain the following information –

 - The fact that the student has successfully completed the project as per the syllabus and that it forms a part of the requirements for completing the BSc degree in computer science of University of Mumbai.
 - The name of the student and the project guide
 - The academic year in which the project is done
 - Date of submission,
 - Signature of the project guide and the head of the department with date along with the department stamp,
 - h. Self-attested copy of Plagiarism Report from any open source tool.
 - i. Index Page detailing description of the following with their subsections:
 - Title: A suitable title giving the idea about what work is proposed.
 - Introduction: An introduction to the topic giving proper background of the topic.
 - Requirement Specification: Specify Software/hardware/data requirements.
 - System Design details : Methodology /Architecture/ UML /DFD etc. used (whichever is applicable)
 - System Implementation: Code implementation
 - Results: Test Cases/Tables/Figures/Graphs/Screenshots/Reports etc.
 - Conclusion and Future Scope: Specify the Final conclusion and future scope
 - References: Books, web links, research articles, etc.
8. The size of the project report shall be around ten to fifteen five pages, excluding the code.
9. The Project report should be submitted in a spiral bound form
10. The Project should be certified by the concerned Project guide and Head of the department.
11. A learner has to make a presentation of working project and will be evaluated as per the Project evaluation scheme given below:
 - a. Presentation of the project : 10 Marks
 - b. Working of the project : 10 Marks
 - c. Quality of the project : 10 Marks
 - d. Viva voce-10 Marks
 - e. Documentation : 10 Marks

(Project will not be accepted without a Certified Project Document at the time of Project Presentation)

CIA-2 for Semester 3

Paper	301	302	303	304	305	306	307
Assignment type	Group Presentation on conceptual topics	Programming Questions	Theoretical Questions and numericals	Programming Questions	Mathematical and conceptual Query Solving	Simple research oriented	Programming Questions
Skill assessed	Team Communication, Leadership qualities, Public Speaking	Programming, Analytical	Theoretical knowledge and its application	Logical Skills and Critical thinking	Problem Solving and critical thinking	Practical implementation of testing software	Practical implementation on visual studio.
Topics							
a	Topics based on Unit I	Unit- I based programs	Questions based on Unit I	Programs based on concepts of Unit I	Queries based on Unit I	Based on Unit-I including SDLC and testing levels	Exploring the controls of ASP.NET
b	Topics based on Unit II	Unit-II based programs	Questions based on Unit II	Programs based on concepts of Unit II	Queries based on Unit II	Unit -II based concepts .	Working with ADO.NET and Ajax

c	Topics based on Unit III	Unit-III based Mini project	Questions based on Unit III	Programs based on concepts of Unit III	Queries based on Unit III	Presentation based on any topic from Unit -III	Presentations topics based on Unit -I
d							
e							

CIA-2 for Semester 4

Paper	401	402	403	404	405	406	407
Assignment type	Query Solving, Db Programming	Programming Questions	Demonstrations on Basic communication concepts of computer networks.	Programming on Raspberry Pi with the help of kit	Mathematical based questions	Practical challenging Questions	Programming Questions
Skill assessed	Database Skill, Critical thinking	Programming, Analytical	Programming on various components of computer	Programming on different IOT topics	Programming based questions	Practical implementation on different concepts	Programming, Analytical

			network ing.				
Topics							
a	Queries based on Unit-I	Programs based on Unit-I	Demonstrations based on Unit-I	Programs based on Unit-I	Practical implementations based on Unit-I	Practical implementations based on Unit-I	Programming Questions
b	Queries based on Unit-II	Programs based on Unit-II	Demonstrations based on Unit-II	Programs based on Unit-II	Practical implementations based on Unit-II	Practical implementations based on Unit-II	Programming Questions
c	Queries based on Unit-III	Mini Project based on Unit-III	Demonstrations based on Unit-III	Presentation based on any topic of Unit-III	Practical implementations based on Unit-III	Practical implementations based on Unit-III	Assignment on theoretical concepts
d							Group assignment based on sample projects

e							Group assignment based on sample projects
---	--	--	--	--	--	--	---

Modality of Assessment

Theory Examination Pattern:

A) Internal Assessment - 40% - 40 Marks

SR. NO	EVALUATION TYPE	MARKS
1	CONTINUOUS INTERNAL ASSESSMENT-I (CIA-I): Internal Class Test with Objective types questions MCQs.	20
2	CONTINUOUS INTERNAL ASSESSMENT-II (CIA-II): Assignments, Case study, Mini project, Group activity, Presentations, Tutorial, Quizzes etc.	20
	TOTAL	40

B) External Examination- 60%- 60 Marks

Semester End Theory Examination: 60 marks

1. **Duration:** These examinations shall be of **2 hours** duration.
2. **Passing criteria:** Students have to acquire a minimum of 40% marks in each course.

3. **Paper Pattern:**

- A. There shall be **four** questions each of **15** marks. On each unit there will be **seven/eight** questions.
- B. All questions shall be compulsory with internal choice within some questions. Each Question may be subdivided into sub questions as a, b, c, d & e, etc & the allocation of Marks` depends on the weightage of the topic

In case if exams are conducted online then the following examination pattern will be followed.

Theory Examination Pattern:

A) Internal Assessment- 40%- 40 Marks

SR. NO	EVALUATION TYPE	MARKS
1	CONTINUOUS INTERNAL ASSESSMENT-I (CIA-I): Internal Class Test with Objective types questions MCQs. Online platforms such as Google Classroom, Google Form, zoom, google meet etc. can be used.	20
2	CONTINUOUS INTERNAL ASSESSMENT-II (CIA-II): Assignments, Case study, Mini project, Group activity, Presentations, Tutorial and Quizzes etc. Online platforms such as Google Classroom, zoom, google meet etc. can be used.	20
	TOTAL	40

**B) External Examination- 60%- 60 Marks Semester End
Theory Examination: 60 marks**

- 1. **Duration:** These examinations shall be of **90mins** duration.
- 2. **Passing criteria:** Students have to acquire a minimum of 40% marks in each course.

3. **Paper Pattern: All MCQ questions.**

Pattern of MCQ question paper should be as follows:

Unit I -14 one mark and 3 two mark questions	20 Marks
Unit II -13 one mark and 3 two mark questions	19 Marks
Unit III - 13 one mark and 4 two mark questions	21 Marks
TOTAL	60 Marks

Practical Examination Pattern:

External Examination- 50 marks

1. **Duration:** These examinations shall be of **2 hours** duration.
2. **Passing criteria:** Students have to acquire a minimum of 40% marks in each course. Minimum 75% practical from each core/allied course are required to be completed and written in the journal.
3. **Certified E-Journal is compulsory for appearing at the time of Practical Exam.**
4. **Examination pattern:**
 - a. Practical work and Viva: 40 marks
 - b. Journal: 10 marks

Overall Examination & Marks Distribution Pattern

Semester III

COURSE	THEORY:	BH.USCS 301,302,303,304,305,306	
	PRACTICALS:	BH.USCSP 301,302,303,304,305,306	
	Internal	External	Total
Theory	40	60	700
Practical's	-	50	300
Total			1000

Semester IV

COURSE	THEORY:	BH.USCS 401,402,403,404,405,406	
	PRACTICALS:	BH.USCSP 401,402,403,404,405,406	
	Internal	External	Total
Theory	40	60	700
Practical's	-	50	300
Total			1000

Rubrics of evaluation for ESE:

Unit	Knowledge	Understanding	Analysis & critical thinking	Total marks/unit
from all units	05	05	5	15
1	05	05	5	15
2	05	05	5	15
3	05	05	5	15
Total	20	20	24	60
% Weightage	33.33	33.33	33.34	100

Rubrics of evaluation for CIA-2 Assignment: Presentation/debate/quiz

Parameters	Max Marks	80 – 100% Excellent	60 -80% Good	40 – 60% Satisfactory	20 – 40% Poor	0-20% very poor
CONTENT	10					
Content: Knowledge	02					
Content: Development	03					
Content: Conclusion	03					

Content: Bibliography	02					
Effective communication skill	10					
Language, Style and Structure;	05					
Teaching aids;	05					
Total	20					

-----X-----