DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW

Evaluation Scheme & Syllabus

For

B.Tech. 2nd Year

Computer Science --- Hindi

(Effective from the Session: 2023-24)

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW

SEMESTER –III

SN	Subject Code	Subject	Type	Category	Periods			Sessional Component		Sessional (SW) (TS/PS)	End Semester Examination (ESE)	Total SW+ESE	Credit Cr
					L	T	P	CT	TA	CT+TA	TE/PE		
1	BOE3**H/ BAS303H	Science Based Open Elective/BSC (Maths-III/Math IV/ Math V)	T	ES/BS	3	1	0	20	10	30	70	100	4
2	BVE301H / BAS301H	Universal Human Value and Professional Ethics/ Technical Communication	T	VA/HS	2	1	0	20	10	30	70	100	3
3	BCS301H	Data Structure	T	PC	3	1	0	20	10	30	70	100	4
4	BCS302H	Computer Organization and Architecture	T	PC	3	1	0	20	10	30	70	100	4
5	BCS303H	Discrete Structures & Theory of Logic	T	PC	2	1	0	20	10	30	70	100	3
6	BCS351H	Data Structure Lab	P	PC	0	0	2		50	50	50	100	1
7	BCS352H	Computer Organization and Architecture Lab	P	PC	0	0	2		50	50	50	100	1
8	BCS353H	Web Designing Workshop	P	PC	0	0	2		50	50	50	100	1
10	BCC301H / BCC302H	Cyber Security/Python programming	T	VA	2	0	0	20	10	30	70	100	2
11	BCC351H	Internship Assessment /Mini Project*	P							100		100	2
		Total			**15**	**5**	**6**						**25**

- **Mathematics –III** for CE / ENV and allied branches
- **Mathematics-IV** for Computer/Electronics/Electrical & allied Branches, Mechanical & Allied Branches Textile/Chemical & allied Branches
- **Mathematics-V** for Bio Technology / Agriculture Engineering

SEMESTER –IV

SN	Subject Code	Subject	Type	Category	Periods			Sessional Component		Sessional (SW) (TS/PS)	End Semester Examination (ESE)	Total SW+ESE	Credit Cr
					L	T	P	CT	TA	CT+TA	TE/PE		
1	BAS403H / BOE4**H	BSC(Maths-III/Math IV/ Math V)/Science Based Open Elective	T	BS/ES	3	1	0	20	10	30	70	100	4
2	BAS401H/ BVE401H	Technical Communication / Universal Human Value and Professional Ethics	T	HS/VA	2	1	0	20	10	30	70	100	3
3	BCS401H	Operating System	T	PC	3	1	0	20	10	30	70	100	4
4	BCS402H	Theory of Automata and Formal Languages	T	PC	3	1	0	20	10	30	70	100	4
5	BCS403H	Object Oriented Programming with Java	T	PC	2	1	0	20	10	30	70	100	3
6	BCS451H	Operating System Lab	P	PC	0	0	2		50	50	50	100	1
7	BCS452H	Object Oriented Programming with Java Lab	P	PC	0	0	2		50	50	50	100	1
8	BCS453H	Cyber Security Workshop	P	PC	0	0	2		50	50	50	100	1
9	BCC402H / BCC401H	Python Programming/Cyber Security	P	VA	2	0	0	20	10	30	70	100	2
10	BVE451H / BVE452H	Sports and Yoga - II / NSS-II	P	VA	0	0	3			100		100	0
		Total			**15**	**5**	**9**						**23**
		Minor Degree/ Honors Degree MT-1/HT-1											

*****The Mini Project or internship (4 weeks) will be done during summer break after 4th Semester and will be assessed during V semester.

SYLLABUS

BCS301H	DATA STRUCTURE	
	Course Outcome (CO)	**Bloom's Knowledge Level (KL)**
	At the end of course , the student will be able to understand	
CO 1	Describe how arrays, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.	**K_1, K_2**
CO 2	Discuss the computational efficiency of the sorting and searching algorithms.	**K_2**
CO 3	Implementation of Trees and Graphs and perform various operations on these data structure.	**K_3**
CO 4	Understanding the concept of recursion, application of recursion and its implementation and removal of recursion.	**K_4**
CO 5	Identify the alternative implementations of data structures with respect to its performance to solve a real world problem.	**K_5, K_6**
	DETAILED SYLLABUS	**3-1-0**

Unit	Topic	Proposed Lecture
I	**Introduction:** Basic Terminology, Elementary Data Organization, Built in Data Types in C. Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big Oh, Big Theta and Big Omega, Time-Space trade-off. Abstract Data Types (ADT) **Arrays**: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Derivation of Index Formulae for 1-D,2-D,3-D and n-D Array Application of arrays, Sparse Matrices and their representations. **Linked lists:** Array Implementation and Pointer Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition Subtraction & Multiplications of Single variable & Two variables Polynomial.	08
II	**Stacks:** Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Iteration and Recursion- Principles of recursion, Tail recursion, Removal of recursion Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers, and Hanoi towers. Tradeoffs between iteration and recursion. **Queues:** Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.	08
III	**Searching:** Concept of Searching, Sequential search, Index Sequential Search, Binary Search. Concept of Hashing & Collision resolution Techniques used in Hashing. **Sorting:** Insertion Sort, Selection, Bubble Sort, Quick Sort, Merge Sort, Heap Sort and Radix Sort.	08
IV	Trees: Basic terminology used with Tree, Binary Trees, Binary Tree Representation: Array Representation and Pointer(Linked List) Representation, Binary Search Tree, Strictly Binary Tree ,Complete Binary Tree . A Extended Binary Trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Constructing Binary Tree from given Tree Traversal, Operation of Insertation , Deletion, Searching & Modification of data in Binary Search .	08

	Threaded Binary trees, Traversing Threaded Binary trees. Huffman coding using Binary Tree. Concept & Basic Operations for AVL Tree , B Tree & Binary Heaps	
V	**Graphs:** Terminology used with Graph, Data Structure for Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.	08

Text books:

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI Learning Private Limited, Delhi India.
2. Gilberg ,Forouzan, Data Structures: A Pseudocode Approach with C 3rd edition , Cengage Learning publication.
3. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
4. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
5. Thareja, "Data Structure Using C" Oxford Higher Education.
6. AK Sharma, "Data Structure Using C", Pearson Education India.
7. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
8. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.
9. P. S. Deshpandey, "C and Data structure", Wiley Dreamtech Publication.
10. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education.
11. Berztiss, AT: Data structures, Theory and Practice, Academic Press.
12. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
13. Adam Drozdek "Data Structures and Algorithm in Java", Cengage Learning

BCS302H	COMPUTER ORGANIZATION AND ARCHITECTURE	
Course Outcome (CO)		**Bloom's Knowledge Level (KL)**
At the end of course , the student will be able to understand		
CO 1	Study of the basic structure and operation of a digital computer system.	K_1, K_2
CO 2	Analysis of the design of arithmetic & logic unit and understanding of the fixed point and floating-point arithmetic operations.	K_2, K_4
CO 3	Implementation of control unit techniques and the concept of Pipelining	K_3
CO 4	Understanding the hierarchical memory system, cache memories and virtual memory	K_2
CO 5	Understanding the different ways of communicating with I/O devices and standard I/O interfaces	K_2, K_4

DETAILED SYLLABUS		**3-1-0**
Unit	**Topic**	**Proposed Lecture**
I	**Introduction**: Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer. Processor organization, general registers organization, stack organization and addressing modes.	**08**
II	**Arithmetic and logic unit:** Look ahead carries adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit design. IEEE Standard for Floating Point Numbers	**08**
III	**Control Unit:** Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc), micro operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Pipelining. Hardwire and micro programmed control: micro programme sequencing, concept of horizontal and vertical microprogramming.	**08**
IV	**Memory:** Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues & performance, address mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation.	**08**
V	**Input / Output**: Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Serial Communication: Synchronous & asynchronous communication, standard communication interfaces.	**08**

Text books:

1. Computer System Architecture - M. Mano

2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012

3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998. Reference books

4. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.

5. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.

6. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012

7. Structured Computer Organization, Tannenbaum(PHI)

BCS303H	Discrete Structures & Theory of Logic	
	Course Outcome (CO)	**Bloom's Knowledge Level (KL)**
	At the end of course , the student will be able to understand	
CO 1	Acquire Knowledge of sets and relations for solving the problems of POSET and lattices.	K_3, K_4
CO 2	Apply fundamental concepts of functions and Boolean algebra for solving the problems of logical abilities.	K_1, K_2
CO 3	Employ the rules of propositions and predicate logic to solve the complex and logical problems.	K_3
CO 4	Explore the concepts of group theory and their applications for solving the advance technological problems.	K_1, K_4
CO 5	Illustrate the principles and concepts of graph theory for solving problems related to computer science.	K_2, K_6

DETAILED SYLLABUS		**3-1-0**
Unit	**Topic**	**Proposed Lecture**
I	**Set Theory& Relations**: Introduction, Combination of sets. Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Recursive definition of relation, Order of relations. **POSET & Lattices**: Hasse Diagram, POSET, Definition & Properties of lattices – Bounded, Complemented, Distributed, Modular and Complete lattice.	**08**
II	**Functions**: Definition, Classification of functions, Operations on functions. Growth of Functions. **Boolean Algebra:** Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps.	**08**
III	**Theory of Logics**: Proposition, Truth tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference. Predicate Logic: First order predicate, well-formed formula of predicate, quantifiers, Inference theory of predicate logic.	**08**
IV	**Algebraic Structures**: Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphisms, Definition and elementary properties of Rings and Fields.	**08**
V	**Graphs**: Definition and terminology, Representation of graphs, Multigraphs, Bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring. **Combinatorics**: Introduction, Counting Techniques, Pigeonhole Principle	**08**

Text books:

1.Koshy, Discrete Structures, Elsevier Pub. 2008 Kenneth H. Rosen, Discrete Mathematics and Its Applications, 6/e, McGraw-Hill, 2006.

2. B. Kolman, R.C. Busby, and S.C. Ross, Discrete Mathematical Structures, 5/e, Prentice Hall, 2004.

3.E.R. Scheinerman, Mathematics: A Discrete Introduction, Brooks/Cole, 2000.

4.R.P. Grimaldi, Discrete and Combinatorial Mathematics, 5/e, Addison Wesley, 2004

5.Liptschutz, Seymour, " Discrete Mathematics", McGraw Hill.

6.Trembley, J.P & R. Manohar, "Discrete Mathematical Structure with Application to Computer Science", McGraw Hill. 4. Deo,

7.Narsingh, "Graph Theory With application to Engineering and Computer.Science.", PHI.

8. Krishnamurthy, V., "Combinatorics Theory & Application", East-West Press Pvt. Ltd., New Delhi

BCS351H- Data Structure Lab

List of Experiments (Indicative & not limited to)

1. **Implementing Sorting Techniques:** Bubble Sort, Insertion Sort, Selection Sort, Shell , Sort, Radix Sort, Quick sort
2. **Implementing Searching and Hashing Techniques:** Linear search, Binary search, Methods for Hashing: Modulo Division, Digit Extraction, Fold shift, Fold Boundary, Linear Probe for Collision Resolution. Direct and Subtraction hashing
3. **Implementing Stacks:** Array implementation, Linked List implementation, Evaluation of postfix expression and balancing of parenthesis , Conversion of infix notation to postfix notation
4. **Implementing Queue:** Linked List implementation of ordinary queue, Array implementation of circular queue, Linked List implementation of priority queue, Double ended queue
5. **Implementing Linked List:** Singly Linked Lists, Circular Linked List, Doubly Linked Lists : Insert, Display, Delete, Search, Count, Reverse(SLL), Polynomial , Addition , Comparative study of arrays and linked list
6. **Implementing Trees:** Binary search tree : Create, Recursive traversal: preorder, post order, in order, Search Largest , Node, Smallest Node, Count number of nodes, Heap: Min Heap, Max Heap: reheap Up, reheap Down, Delete , Expression Tree, Heapsort
7. **Implementing Graphs**: Represent a graph using the Adjacency Matrix, BFS, Find the minimum spanning tree (using any method Kruskal's Algorithm or Prim's Algorithm) Self Learning Topics : Shortest Path Algorithm

BCS352H- Computer Organization Lab

List of Experiments (Indicative & not limited to)

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER.
4. Implementing 4x1 and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT.
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.

BCS353H- Web Designing Workshop

Syllabus:

- **HTML:** Elements, attributes, heading, paragraph, styles, comments, links, images, favicon, tables, list, class, id, HTML forms, HTML media, navigation bar.
- **CSS**: Types of CSS, colors, background, margins, padding, height, width, text, font, icon, links, list, tables, display, z-index, float, overflow, CSS media queries, inline block, navigation bar, image gallery, forms, round corners
- **BOOTSTRAP** : Fundamentals of implementing responsive web design ,Use Balsamiq to mockup and wireframe websites, The fundamentals of UI design for websites ,How to install the Bootstrap framework ,Understanding the Bootstrap grid layout system, How to use bootstrap containers to layout your website easily, Use other Bootstrap components such as buttons ,Adding symbols using Font Awesome, Bootstrap carousels. Add Bootstrap cards to your website. Using Bootstrap navigation bars,
- **JavaScript** script, function, output, statement, variables, operators, datatypes, objects, events, string methods, Arrays, if else, switch, loop for, loop in, loop for, debugging, validation of forms , Functions and invocation patterns Discussion of ECMAScripts Intermediate JavaScript , JS Expressions, Operators, Statements and Declarations , Object-Oriented Programming JS Objects and Prototypes, `This`, Scope and Closures Objects and Prototypes Refactoring and Debugging

Textbook

1. Meloni, J. C., Kyrnin, J. (2018). HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself. United Kingdom: Pearson Education.
2. McGrath, M. (2020). HTML, CSS & JavaScript in easy steps. United Kingdom: In Easy Steps Limited.

Reference Books

1. Duckett, J. (2014). Web Design with HTML, CSS, JavaScript and JQuery Set. United Kingdom: Wiley.
2. Fajfar, I. (2015). Start Programming Using HTML, CSS, and JavaScript. United Kingdom: CRC Press.

List of Experiments (Indicative & not limited to)

Experiment No.	List of Experiments (Indicative & not limited to)
1	Designthefollowingstaticwebpagesrequiredforanonlinebookstorewebsite. **HOMEPAGE:** - The static home page must contain three **frames**. - Top frame: Logo and the college name and links to Homepage, Login

page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

For example: When you click the link **"CSE"** the catalogue for **CSE** Books should be displayed in the Right frame. Right frame: The *pages to the links in the left frame must be loaded here*. Initially this page contains description of the web site.

Logo	**Web Site Name**			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Description of the WebSite			

2.

LOGINPAGE:

This page looks like below:

Logo	WebSite Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Login Page User Name: Passwords: Submit Reset			

3.

CATOLOGUE PAGE: The catalogue page should contain the details of all the books available in the website in a table. The details should contain the following:

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

Logo	WebSite Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE	  	Book:XMLBible Author : Winston Publication:Wiely	$40.5	Add to cart

		EEE CIVIL		Book :AI Author:S.Russel Publication:Princetonhall Book : Java 2 Author:Watson Publication:BPBpublications Book : HTML in 24 hours Author : Sam Peter Publication:Sampublication	$63 $35.5 $50	Add to cart Add to cart Add to cart

4.

CARTPAGE: The cart page contains the details about the books which are added to the cart. The cart page should look like this:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE	Book name	Price	Quantity	Amount
ECE	Java 2	$35.5	2	$70
EEE	XML bible	$40.5	1	$40.5
CIVIL	Total amount	- $130.5		

5.

REGISTRATION PAGE : Create a" registration form" with the following fields

1)Name (Text field)

2)Password (password field)

3) E-mailid(text field)
4) Phone Number(text field)
5) Sex(radio button)
6) Date of birth(3 select boxes)
7) Languages known(checkboxes–English, Telugu, Hindi, Tamil)
8) Address(text area)

6.

Js VALIDATION: Write ***JavaScript*** **to validate** the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).

7.

Js VALIDATION:

3. E-mailid (should not contain any invalid and must follow the standard pattern(name@domain.com)
4. Phone Number(Phone number should contain 10 digits only).

8.

CSS: Design a web page using **CSS(C**ascading **S**tyle **S**heets**)** which includes the following:

1) Use different font, styles:
In the style definition you define how each selector should work(font, color etc.). Then, in the body of your pages, you refer to these selectors to activate the styles.

2) Set a background image for both the page and single elements on the page.

9.	**CSS:** 1) Control the repetition of the image with the background-repeat property. 2) Define styles for links as A:link A:visited A:active A:hover
10.	Consider a small topic of your choice on which you can develop static Webpages and try to implement all topics of html, CSS and Js within the topic. Choose any one topic. 1. Your Own Portfolio 2. To-Do List 3. Survey Form 4. A Tribute Page 5. A Questionnaire

FOURTH SEMESTER (DETAILED SYLLABUS)

BCS401H	Operating system	
Course Outcome (CO)		**Bloom's Knowledge Level (KL)**
At the end of course , the student will be able to understand		
CO 1	Understand the structure and functions of OS	K_1, K_2
CO 2	Learn about Processes, Threads and Scheduling algorithms.	K_1, K_2
CO 3	Understand the principles of concurrency and Deadlocks	K_2
CO 4	Learn various memory management scheme	K_2
CO 5	Study I/O management and File systems.	K_2,K_4
DETAILED SYLLABUS		**3-0-0**

Unit	Topic	Proposed Lecture
I	**Introduction** : Operating system and functions, Classification of Operating systems- Batch, Interactive, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.	**08**
II	**Concurrent Processes**: Process Concept, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.	**08**
III	**CPU Scheduling:** Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Algorithms, Multiprocessor Scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.	**08**
IV	**Memory Management:** Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.	**08**
V	**I/O Management and Disk Scheduling**: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, File system implementation issues, File system protection and security.	**08**

Text books:

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley
2. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education
3. Harvey M Dietel, " An Introduction to Operating System", Pearson Education
4. D M Dhamdhere, "Operating Systems : A Concept based Approach", 2nd Edition,

5. TMH 5. William Stallings, “Operating Systems: Internals and Design Principles ”, 6th Edition, Pearson Education

BCS402H	Theory of Automata and Formal Languages	
	Course Outcome (CO)	**Bloom's Knowledge Level (KL)**
	At the end of course , the student will be able to understand	
CO 1	Analyse and design finite automata, pushdown automata, Turing machines, formal languages, and grammars	K_4, K_6
CO 2	Analyse and design, Turing machines, formal languages, and grammars	K_4, K_6
CO 3	Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving	K_1, K_5
CO 4	Prove the basic results of the Theory of Computation.	K_2,K_3
CO 5	State and explain the relevance of the Church-Turing thesis.	K_1, K_5

DETAILED SYLLABUS		**3-1-0**
Unit	**Topic**	**Proposed Lecture**
I	**Basic Concepts and Automata Theory:** Introduction to Theory of Computation- Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore Machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata.	**08**
II	**Regular Expressions and Languages:** Regular Expressions, Transition Graph, Kleen's Theorem, Finite Automata and Regular Expression- Arden's theorem, Algebraic Method Using Arden's Theorem, Regular and Non-Regular Languages- Closure properties of Regular Languages, Pigeonhole Principle, Pumping Lemma, Application of Pumping Lemma, Decidability- Decision properties, Finite Automata and Regular Languages	**08**
III	**Regular and Non-Regular Grammars**: Context Free Grammar(CFG)-Definition, Derivations, Languages, Derivation Trees and Ambiguity, Regular Grammars-Right Linear and Left Linear grammars, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms- Chomsky Normal Form(CNF), Greibach Normal Form (GNF), Chomsky Hierarchy, Programming problems based on the properties of CFGs.	**08**
IV	**Push Down Automata and Properties of Context Free Languages**: Nondeterministic Pushdown Automata (NPDA)- Definition, Moves, A Language Accepted by NPDA, Deterministic Pushdown Automata(DPDA) and Deterministic Context free Languages(DCFL), Pushdown Automata for Context Free Languages, Context Free grammars for Pushdown Automata, Two stack Pushdown Automata, Pumping Lemma for CFL, Closure properties of CFL, Decision Problems of CFL, Programming problems based on the properties of CFLs.	**08**
V	**Turing Machines and Recursive Function Theory** : Basic Turing Machine Model, Representation of Turing Machines, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Turing Machine as Computer of Integer Functions, Universal Turing machine, Linear Bounded Automata, Church's Thesis, Recursive and Recursively Enumerable language, Halting Problem, Post's Correspondance Problem, Introduction to Recursive Function Theory.	**08**

Text books:

1. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia
2. Introduction to languages and the theory of computation, J Martin, 3rd Edition, Tata McGraw Hill
3. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI

4. Mathematical Foundation of Computer Science, Y.N.Singh, New Age Internationa

BCS403H **Object Oriented Programming with Java**

Course Outcome (CO)		Bloom's Knowledge Level (KL)
At the end of course , the student will be able to understand		
CO 1	Develop the object-oriented programming concepts using Java	K_3, K_4
CO 2	Implement exception handling, file handling, and multi-threading in Java	K_2,K_4
CO 3	Apply new java features to build java programs.	K_3
CO 4	Analyse java programs with Collection Framework	K_4
CO 5	Test web and RESTful Web Services with Spring Boot using Spring Framework concepts	K_5

DETAILED SYLLABUS		**3-1-0**
Unit	**Topic**	**Proposed Lecture**
I	**Introduction:** Why Java, History of Java, JVM, JRE, Java Environment, Java Source File Structure, and Compilation. Fundamental, **Programming Structures in Java:** Defining Classes in Java, Constructors, Methods, Access Specifies, Static Members, Final Members, Comments, Data types, Variables, Operators, Control Flow, Arrays & String. **Object Oriented Programming**: Class, Object, Inheritance Super Class, Sub Class, Overriding, Overloading, Encapsulation, Polymorphism, Abstraction, Interfaces, and Abstract Class. **Packages:** Defining Package, CLASSPATH Setting for Packages, Making JAR Files for Library Packages, Import and Static Import Naming Convention For Packagesprocessor evolution and types, microprocessor architecture and operation of its components, addressing modes, interrupts, data transfer schemes, instruction and data flow, timer and timing diagram, Interfacing devices.	**08**
II	**Exception Handling**: The Idea behind Exception, Exceptions & Errors, Types of Exception, Control Flow in Exceptions, JVM Reaction to Exceptions, Use of try, catch, finally, throw, throws in Exception Handling, In-built and User Defined Exceptions, Checked and Un-Checked Exceptions. **Input /Output Basics**: Byte Streams and Character Streams, Reading and Writing File in Java. **Multithreading:** Thread, Thread Life Cycle, Creating Threads, Thread Priorities, Synchronizing Threads, Inter-thread Communication.	**08**
III	**Java New Features:** Functional Interfaces, Lambda Expression, Method References, Stream API, Default Methods, Static Method, Base64 Encode and Decode, ForEach Method, Try-with-resources, Type Annotations, Repeating Annotations, Java Module System, Diamond Syntax with	**08**

	Inner Anonymous Class, Local Variable Type Inference, Switch Expressions, Yield Keyword, Text Blocks, Records, Sealed Classes	
IV	**Java Collections Framework:** Collection in Java, Collection Framework in Java, Hierarchy of Collection Framework, Iterator Interface, Collection Interface, List Interface, ArrayList, LinkedList, Vector, Stack, Queue Interface, Set Interface, HashSet, LinkedHashSet, SortedSet Interface, TreeSet, Map Interface, HashMap Class, LinkedHashMap Class, TreeMap Class, Hashtable Class, Sorting, Comparable Interface, Comparator Interface, Properties Class in Java.	**08**
V	**Spring Framework:** Spring Core Basics-Spring Dependency Injection concepts, Spring Inversion of Control, AOP, Bean Scopes- Singleton, Prototype, Request, Session, Application, Web Socket, Auto wiring, Annotations, Life Cycle Call backs, Bean Configuration styles **Spring Boot:** Spring Boot Build Systems, Spring Boot Code Structure, Spring Boot Runners, Logger, BUILDING RESTFUL WEB SERVICES, Rest Controller, Request Mapping, Request Body, Path Variable, Request Parameter, GET, POST, PUT, DELETE APIs, Build Web Applications	**08**

Text Books

1. Herbert Schildt, "Java The complete reference", McGraw Hill Education
2. Craig Walls, “Spring Boot in Action” Manning Publication
1. Steven Holzner, “Java Black Book”, Dreamtech.
2. Balagurusamy E, “Programming in Java”, McGraw Hill
3. Java: A Beginner’s Guide by Herbert Schildt, Oracle Press
4. Greg L. Turnquist “Learning Spring Boot 2.0 - Second Edition”, Packt Publication
5. AJ Henley Jr (Author), Dave Wolf, “Introduction to Java Spring Boot: Learning by Coding”, Independently Published

BCS451H- Operating System Lab

List of Experiments (Indicative & not limited to)

1. Study of hardware and software requirements of different operating systems (UNIX,LINUX,WINDOWS XP, WINDOWS7/8
2. Execute various UNIX system calls for
    i. Process management
    ii. File management
    iii. Input/output Systems calls
3. Implement CPU Scheduling Policies:
    i. SJF
    ii. Priority
    iii. FCFS
    iv. Multi-level Queue
4. Implement file storage allocation technique:
    i. Contiguous(using array)

ii. Linked –list(using linked-list)
iii. Indirect allocation (indexing)

5. Implementation of contiguous allocation techniques:
   i. Worst-Fit
   ii. Best- Fit
   iii. First- Fit
6. Calculation of external and internal fragmentation
   i. Free space list of blocks from system
   ii. List process file from the system
7. Implementation of compaction for the continually changing memory layout and calculate total movement of data
8. Implementation of resource allocation graph RAG)
9. Implementation of Banker‟s algorithm
10. Conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
11. Implement the solution for Bounded Buffer (producer-consumer)problem using inter process communication techniques-Semaphores
12. Implement the solutions for Readers-Writers problem using inter process communication technique - Semaphore

BCS452H- Object Oriented Programming with Java

List of Experiments (Indicative & not limited to)

1. Use Java compiler and eclipse platform to write and execute java program.
2. Creating simple java programs using command line arguments
3. Understand OOP concepts and basics of Java programming.
4. Create Java programs using inheritance and polymorphism.
5. Implement error-handling techniques using exception handling and multithreading.
6. Create java program with the use of java packages.
7. Construct java program using Java I/O package.
8. Create industry oriented application using Spring Framework.
9. Test RESTful web services using Spring Boot.
10. Test Frontend web application with Spring Boot

BCS453H- Cyber Security Workshop

List of Experiments (Indicative & not limited to)

Module 1: Packet Analysis using Wire shark

1. Basic Packet Inspection: Capture network traffic using Wire shark and analyze basic protocols like HTTP, DNS, and SMTP to understand how data is transmitted and received.

2. Detecting Suspicious Activity: Analyze network traffic to identify suspicious patterns, such as repeated connection attempts or unusual communication between hosts.
3. Malware Traffic Analysis: Analyze captured traffic to identify signs of malware communication, such as command-and-control traffic or data infiltration.
4. Password Sniffing: Simulate a scenario where a password is transmitted in plaintext. Use Wireshark to capture and analyze the packets to demonstrate the vulnerability and the importance of encryption.
5. ARP Poisoning Attack: Set up an ARP poisoning attack using tools like Ettercap. Analyze the captured packets to understand how the attack can lead to a Man-in-the-Middle scenario.

Module 2: Web Application Security using DVWA

1. SQL Injection: Use DVWA to practice SQL injection attacks. Demonstrate how an attacker can manipulate input fields to extract, modify, or delete database information.
2. Cross-Site Scripting (XSS): Exploit XSS vulnerabilities in DVWA to inject malicious scripts into web pages. Show the potential impact of XSS attacks, such as stealing cookies or defacing websites.
3. Cross-Site Request Forgery (CSRF): Set up a CSRF attack in DVWA to demonstrate how attackers can manipulate authenticated users into performing unintended actions.
4. File Inclusion Vulnerabilities: Explore remote and local file inclusion vulnerabilities in DVWA. Show how attackers can include malicious files on a server and execute arbitrary code.
5. Brute-Force and Dictionary Attacks: Use DVWA to simulate login pages and demonstrate brute-force and dictionary attacks against weak passwords. Emphasize the importance of strong password policies.