Draft of the Scheme, Syllabus and Course Plan of B-Tech Computer Science & Engineering (S3 & S4) of APJ Abdul Kalam Kerala Technological University

(Not for circulation)

SEMESTER - 3				
Course Code	***Course***	***L-T-P***	***Credits***	***Exam Slot***
~~MA210~~	~~Linear Algebra~~	~~3-1-0~~	~~4~~	~~A~~
CS201	Discrete Computational Structures	3-1-0	4	B
CS203	Switching Theory and Logic Design	3-1-0	4	C
CS205	Data Structures and Algorithms	3-1-0	4	D
CS209	Electronic Devices & Circuits	3-0-0	3	F
CS231	Data Structures Lab	0-0-3	1	S
CS233	Electronics Circuits Lab	0-0-3	1	T
~~HS310~~	~~Life Skills~~	~~2-0-2~~	~~3~~	~~E~~
	Total Credits = 24 Hours : 28/29			
	Cumulative Credits = 71			

Course No.	Course Name	L-T-P Credits	Year of Introduction
CS201	**DISCRETE COMPUTATIONAL STRUCTURES**	**3-1-0-4**	**2015**

Course Objectives

1. To introduce mathematical notations and concepts in discrete mathematics that are essential for computing.
2. To train on mathematical reasoning and proof strategies.
3. To impart fundamental concepts of Discrete Mathematical Structures.
4. To cultivate analytical thinking and creative problem solving skills.

Syllabus

Review of Set theory, Countable and uncountable Sets, Prepositional and Predicate Calculus, Proof Techniques, Algebraic systems (semigroups, monoids, groups, rings, fields, homomorphism), Posets and Lattices, Review of Permutations and combinations, Pigeon Hole Principle, Recurrence Relations and Solutions

Course Outcome:

Student is able to

1. Perform operations on discrete structures such as sets, relations and functions
2. Verify the validity of an argument using propositional logic
3. Verify the validity of an argument using predicate logic
4. Construct proofs using direct proof, proof by contraposition, proof by contradiction and proof by cases, and mathematical induction.
5. Solve problems using algebraic structures.
6. Solve problems using counting techniques and combinatorics
7. Solve problems involving recurrence relations.

Text Books

1. Trembly J.P and Manohar R, “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw–Hill Pub.Co.Ltd, New Delhi, 2003.
2. Liu C. L., “Elements of Discrete Mathematics”, 2/e, McGraw–Hill Int. editions, 1988.

References:

1. Ralph. P. Grimaldi, “Discrete and Combinatorial Mathematics: An Applied Introduction”, 4/e, Pearson Education Asia, Delhi, 2002.
2. Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, “Discrete Mathematical Structures”, Pearson Education Pvt Ltd., New Delhi, 2003
3. Kenneth H.Rosen, “Discrete Mathematics and its Applications”, 5/e, Tata McGraw – Hill Pub. Co. Ltd., New Delhi, 2003.
4. Richard Johnsonbaugh, “Discrete Mathematics”, 5/e, Pearson Education Asia, New Delhi, 2002.
5. Joe L Mott, Abraham Kandel, Theodore P Baker, “Discrete Mathematics for Computer Scientists and Mathematicians”, 2/e, Prentice-Hall India, 2009.

Course Plan			
Module	**Contents**	**Hours (56)**	**End Sem Exam Marks**
I	**Review of elementary set theory** : Algebra of sets – Ordered pairs and Cartesian products – Countable and Uncountable sets	3	15 %
	Relations :- Relations on sets –Types of relations and their properties – Relational matrix and the graph of a relation – Partitions – Equivalence relations - Partial ordering- Posets – Hasse diagrams - Meet and Join – Infimum and Supremum	6	
	Functions *:-* *Injective, Surjective and Bijective functions - Inverse of a function- Composition*	1	
II	**Propositional Logic**:- Propositions – Logical connectives – Truth tables	2	15 %
	Tautologies and contradictions – Contra positive – Logical equivalences and implications	3	
	Rules of inference : Validity of arguments.	3	
FIRST INTERNAL EXAM			
III	**Predicate Logic**:- Predicates – Variables – Free and bound variables – Universal and Existential Quantifiers – Universe of discourse	3	15 %
	Logical equivalences and implications for quantified statements – Theory of inference : Validity of arguments.	3	
	Proof techniques: Mathematical induction and its variants – Proof by Contradiction – Proof by Counter Example – Proof by Contra positive.	3	
IV	**Algebraic systems**:- Semigroups and monoids - Homomorphism, Subsemigroups and submonoids	2	15 %
	Groups, definition and elementary properties, subgroups, Homomorphism and Isomorphism, Generators - Cyclic Groups, Cosets and Lagrange's Theorem	5	
	Algebraic systems with two binary operations- rings, fields-sub rings, ring homomorphism	2	
SECOND INTERNAL EXAM			
V	**Lattices and Boolean algebra** :-		20 %

	Lattices – Sublattices – Complete lattices – Bounded Lattices - Complemented Lattices – Distributive Lattices – Lattice Homomorphisms. Boolean algebra – sub algebra, direct product and homomorphisms	6 3	
VI	**Combinatorics:** Review of Permutations, Combinations, Pigeonhole principle and Principle of Inclusion Exclusion. **Recurrence Relations**: Introduction- Linear recurrence relations with constant coefficients– Homogeneous solutions – Particular solutions – Total solutions	3 5	20 %
END SEMESTER EXAM			

Course No.	Course Name	L-T-P-Credits	Year of Introduction
CS203	**Switching Theory and Logic Design**	**3-1-0-4**	**2015**

Course Objectives

1. To impart an understanding of the basic concepts of Boolean algebra and digital systems.
2. To impart familiarity with the design and implementation of different types of practically used sequential circuits.
3. To provide an introduction to use Hardware Description Language

Syllabus

Introduction to Number Systems, Boolean Algebra, Canonical Forms, Logic Gates, Digital Circuit Design, Combination Logic Circuit Design, Sequential Circuit Design, Registers, Counter, Memory modules, Programmable Logical Arrays, Hardware Description Language for Circuit Design, Arithmetic algorithms

Course Outcomes

Student is able to:-

1. Apply the basic concepts of Boolean algebra for the simplification and implementation of logic functions using suitable gates namely NAND, NOR etc.
2. Design simple Combinational Circuits such as Adders, Subtractors, Code Convertors, Decoders, Multiplexers,Magnitude Comparators etc.
3. Analyze and Design simple and commonly used Sequential Circuits viz. different types of Counters, Shift Registers, Serial Adders, Sequence Generators.
4. Use of Programmable Logic Arrays for implementing Boolean functions.
5. Use Hardware Description Language for describing simple logic circuits.
6. Explain and illustrate algorithms for addition/subtraction operations on Binary, BCD and Floating Point Numbers.

Text Books:

1. Mano M. M., *Digital Logic & Computer Design*, 4/e, Pearson Education, 2013. [Chapters: 1, 2, 3, 4, 5, 6, 7].
2. Floyd T. L., *Digital Fundamentals*, 10/e, Pearson Education, 2009. [Chapters: 5, 6].
3. M. Morris Mano, *Computer System Architecture*, 3/e, Pearson Education, 2007. [Chapter 10.1, 10.2, 10.5, 10.6, 10.7].
4. Harris D. M. and, S. L. Harris, Digital *Design and Computer Architecture*, 2/e, Morgan Kaufmann Publishers, 2013 [Chapter 4.1, 4.2]

References:

1. Tokheim R. L., *Digital Electronics Principles and Applications*, 7/e, Tata McGraw Hill, 2007.
2. Mano M. M. and M. D Ciletti, *Digital Design*, 4/e, Pearson Education, 2008.
3. Rajaraman V. and T. Radhakrishnan, *An Introduction to Digital Computer Design*, 5/e, Prentice Hall India Private Limited, 2012.
4. Leach D, Malvino A P, Saha G, *Digital Principles and Applications, 8/e, McGraw Hill Education, 2015.*

COURSE PLAN

Module	Contents	Contact Hours	Sem. Exam Marks;%

		(52)	
I	Number systems – Decimal, Binary, Octal and Hexadecimal – conversion from one system to another – representation of negative numbers – representation of BCD numbers – character representation – character coding schemes – ASCII – EBCDIC etc. Addition, subtraction, multiplication and division of binary numbers (no algorithms). Addition and subtraction of BCD, Octal and Hexadecimal numbers. Representation of floating point numbers – precision – addition, subtraction, multiplication and division of floating point numbers	**10**	**15%**
II	Introduction — Postulates of Boolean algebra – Canonical and Standard Forms — logic functions and gates methods of minimization of logic functions — Karnaugh map method and Quin McClusky method Product-of-Sums Simplification — Don't-Care Conditions.	**09**	**15%**
III	Combinational Logic: combinational Circuits and design Procedure — binary adder and subtractor — multi—level NAND and NOR circuits — Exclusive-OR and Equivalence Functions. Implementation of combination logic: parallel adder, carry look ahead adder, BCD adder, code converter, magnitude comparator, decoder, multiplexer, de-multiplexer, parity generator.	**10**	**15%**
IV	Sequential logic circuits: latches and flip-flops – edge-triggering and level-triggering — RS, JK, D and T flip-flops — race condition — master-slave flip-flop. Clocked sequential circuits: state diagram — state reduction and assignment — design with state equations	**08**	**15%**
	Registers: registers with parallel load - shift registers universal shift registers – application: serial adder.	**08**	**20%**

V	Counters: asynchronous counters — binary and BCD ripple counters — timing sequences — synchronous counters — up-down counter, BCD counter, Johnson counter — timing sequences and state diagrams.		
VI	Memory and Programmable Logic: Random-Access Memory (RAM)—Memory Decoding—Error Detection and Correction — Read only Memory (ROM), Programmable Logic Array (PLA). *HDL: fundamentals, combinational logic, adder, multiplexer.* Arithmetic algorithms: Algorithms for addition and subtraction of binary and BCD numbers, algorithms for floating point addition and subtraction.	**08**	**20%**

Course No.	Course Name	L-T-P-Credits	Year of Introduction
CS205	**Data Structures and Algorithms**	**3-1-0-4**	**2015**

Course Objectives

1. To impart basic techniques of algorithm design and analysis
2. To impart a thorough understanding of linear data structures such as stacks queues and their applications.
3. To impart a thorough understanding of non-linear data structures such as trees, graphs and their applications.
4. To impart familiarity with various sorting, searching and hashing techniques and their performance comparison.
5. To impart a basic understanding of memory management.

Syllabus

Introduction to various programming methodologies, terminologies and basics of algorithms analysis, Basic Abstract and Concrete Linear Data Structures, Non-linear Data Structures, Memory Management, Sorting Algorithms, Searching Algorithms, Hashing.

Course Outcomes

Student is able to:-

1. Compare different programming methodologies and define asymptotic notations to analyze performance of algorithms.
2. Use appropriate data structures like arrays, linked list, stacks and queues to solve real world problems efficiently.
3. Represent and manipulate data using nonlinear data structures like trees and graphs to design algorithms for various applications.
4. Explain, illustrate and compare various techniques for searching sorting and hashing.
5. Explain different memory management techniques and their significance.
6. Explain and illustrate hashing techniques.

Text Books:

1. Horowitz E. and S. Sahni, Fundamentals of Data Structures, Galgotia Publications.
2. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2/e, Cengage Learning, 2005.

References

1. Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983.
2. Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.
3. Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.
4. Lipschuts S., Theory and Problems of Data Structures, Schaum's Series, 1986.
5. Horwitz E., S. Sahni and S. Anderson, Fundamentals of Data Structures in C, University Press (India), 2008.
6. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall, 2004.

7. Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI, 1987.
8. Martin Barrett, Clifford Wagner, And Unix: Tools For Software Design, John Wiley, 2008 reprint.

COURSE PLAN

Module	Contents	Contact Hours (50)	Sem. Exam Marks;%
I	Introduction to programming methodologies – structured approach, stepwise refinement techniques, programming style, documentation – analysis of algorithms: frequency count, definition of Big O notation, asymptotic analysis of simple algorithms. Recursive and iterative algorithms.	8	15%
II	Abstract and Concrete Data Structures- Basic data structures – vectors and arrays Linked lists:- singly linked list, doubly linked list, Circular linked list, operations on linked list, linked list with header nodes, applications of linked list.	8	15%
III	Implementation of Stacks and Queues using arrays and linked list, DEQUEUE (double ended queue). Multiple Stacks and Queues, Applications. String: - representation of strings, concatenation, substring searching and deletion.	8	15%
IV	Trees: - m-ary Tree, Binary Trees – level and height of the tree, complete-binary tree representation using array, tree traversals, applications. Binary search tree – creation, insertion and deletion ans search operations, applications. Graphs – representation of graphs, BFS and DFS (analysis not required)	10	15%
V	Sorting techniques – Bubble sort, Selection Sort, Insertion sort, Merge sort, Quick sort, Heaps and Heap sort. Searching algorithms – Linear and Binary search. (For all the above a basic analysis of the algorithms and performance comparison are expected.)	08	20%
VI	Memory management: - reference count, garbage collection algorithm-algorithm for marking accessible cells. Fragmentation and compaction-first fit, best fit, boundary tag method (basic concepts only; algorithms not expected.)	08	20%

	Hash Tables – Hashing functions – Mid square, division, folding, digit analysis, Overflow handling.		

Course No.	Course Name	L-T-P Credits	Year of Introduction
CS209	**ELECTRONIC DEVICES & CIRCUITS**	**3-0-0-3**	**2015**

Course Objectives:

1. To introduce to the students the fundamental concepts of electronic devices and circuits for engineering applications
2. To develop the skill of analysis and design of various analog circuits using electronic devices
3. To provide comprehensive idea about working principle, operation and applications of electronic circuits
4. To equip the students with a sound understanding of fundamental concepts of operational amplifiers
5. To expose to the diversity of operations that operational amplifiers can perform in a wide range of applications
6. To expose to a variety of electronic circuits/systems using various analog ICs

Syllabus

RC Circuits, Diode Circuits, Regulated power supplies, **Field effect transistor,** DC analysis of BJT, RC Coupled amplifier, MOSFET amplifiers, , Feedback amplifiers, Power amplifiers, Oscillators, Multivibrators, Operational Amplifier and its applications, Timer IC.

Course outcome:

At the end of the course, students will be able to

1. explain, illustrate, and design the different electronic circuits using electronic components
2. design circuits using operational amplifiers for various applications

Text Books:

1. David A Bell, Electronic Devices and Circuits, Oxford University Press, 2008
2. Salivahanan S. and V. S. K. Bhaaskaran, Linear Integrated Circuits, Tata McGraw Hill, 2008

References :

1. Neamen D., Electronic Circuits, Analysis and Design, 3/e, TMH, 2007
2. Robert Boylestad and L Nashelsky, Electronic Devices and Circuit Theory, Pearson.
3. Bogart T. F., Electronic Devices Circuits, 6/e, Pearson, 2012.
4. Maini A. K. and V. Agrawal, Electronic Devices and Circuits, Wiley India, 2011.
5. K.Gopakumar, Design and Analysis of Electronic Circuits, Phasor Books, Kollam, 2013
6. Millman J. and C. Halkias, Integrated Electronics, 2/e, McGraw-Hill, 2010.

Course Plan

Module	Contents	Hours (40)	Sem Exam Marks

1	**Wave shaping circuits:** Sinusoidal and non-sinusoidal wave shapes, Principle and working of RC differentiating and integrating circuits, Conversion of one non-sinusoidal wave shape into another. Clipping circuits - Positive, negative and biased clipper. Clamping circuits - Positive, negative and biased clamper. Voltage multipliers- Voltage doubler and tripler. Simple sweep circuit using transistor as a switch.	5	15%
2	**Regulated power supplies:** Review of simple zener voltage regulator, Shunt and series voltage regulator using transistors, Current limiting and fold back protection, 3 pin regulators-78XX and 79XX, IC 723 and its use as low and high voltage regulators, DC to DC conversion, Circuit/block diagram and working of SMPS.	4	15 %
	Field effect transistors: JFET – Structure, principle of operation and characteristics, Comparison with BJT. MOSFET- Structure, Enhancement and Depletion types, principle of operation and characteristics.	3	
FIRST INTERNAL EXAM			
3	**Amplifiers:** Introduction to transistor biasing, operating point, concept of load line, thermal stability, fixed bias, self bias, voltage divider bias. Classification of amplifiers, low frequency small signal model (hybrid pi) of RC coupled amplifier - voltage gain and frequency response. Multistage amplifiers - effect of cascading on gain and bandwidth. Feedback in amplifiers - Effect of negative feedback on amplifiers, feedback topologies and comparison, current series feedback amplifier. Power amplifiers - Class A, class B and class AB push pull amplifiers. MOSFET Amplifier- Circuit diagram and working of common source MOSFET amplifier.	7	15 %
4	**Oscillators:** Classification, criterion for oscillation, analysis of Wien bridge oscillator, Hartley and Crystal oscillator. Non-sinusoidal oscillators: Astable, monostable and bi-stable multivibrators using transistors (Only design equations and working of circuit are required, Analysis not required).	5	15 %
SECOND INTERNAL EXAM			

5	**Operational amplifiers:** Differential amplifier, characteristics of op-amps(gain, bandwidth, slew rate, CMRR, offset voltage, offset current), comparison of ideal and practical op-amp(IC741), applications of op-amps- scale changer, sign changer, adder/summing amplifier, subtractor, integrator, differentiator, logarithmic amplifier, instrumentation amplifier, concept of analog computation using op-amp based circuits (solution of a second order ordinary differential equation). Schmitt trigger, Wien bridge oscillator, astable and monostable multivibrators.	8	20 %
6	**Integrated circuits:** Active filters – Low pass and high pass (first and second order) active filters using op-amp with gain (No analysis required). D/A and A/D convertors – important specifications, Sample and hold circuit. Binary weighted resistor and R-2R ladder type D/A convertors. Flash, dual slope and successive approximation type A/D convertors. Block diagram and working of Timer IC555, astable and monostable multivibrators using 555.	8	20 %
END SEMESTER EXAM			

Course No.	Course Name	L-T-P - Credits	Year of Introduction
CS231	**DATA STRUCTURES LAB**	**0-0-3-1**	**2015**

Course Objectives

1. To implement basic linear and non-linear data structures and their major operations.
2. To implement applications using these data structures.
3. To implement algorithms for various sorting techniques.

List of Exercises/Experiments :

1. Implementation of Stack and Multiple stacks using one dimensional array.
2. Application problems using stacks: Infix to post fix conversion, postfix and pre-fix evaluation, MAZE problem etc.
3. Implementation of Queue, DEQUEUE and Circular queue using arrays.
4. Implementation of various linked list operations.
5. Implementation of stack, queue and their applications using linked list.
6. Implementation of trees using linked
7. Representation of polynomials using linked list, addition and multiplication of polynomials.
8. Implementation of binary trees using linked lists and arrays- creations, insertion, deletion and traversal.
9. Implementation of binary search trees – creation, insertion, deletion, search
10. Application using trees
11. Representation of graphs and computing various parameters (in degree, out degree etc.) - adjacency list, adjacency matrix.
12. Implementation of BFS, DFS for each representation.
13. Application using graphs.
14. Implementation of searching and sorting algorithms – bubble, insertion, selection, quick (recursive and non-recursive), merge sort (recursive and non-recursive), and heap sort, linear search, binary search.
15. Implementation of hash table using various mapping functions, various collision and

overflow resolving schemes.

16. Implementation of various string operations.

17. Simulation of first-fit, best-fit and worst-fit allocations.

18. Simulation of a basic memeory allocator and garbage collector using doubly linked list.

Course outcome.

Student is able to:

1. Choose appropriate data structure for a given problem
2. Design algorithms to solve real world problems.
3. Compare the performance of various searching and sorting algorithms quantitatively.

Course No.	Course Name	L-T-P - Credits	Year of Introduction
CS233	**ELECTRONICS CIRCUITS LAB**	**0-0-3-1**	**2015**

Course Objectives:

1. To study the working of analog electronic circuits.
2. To design, implement and demonstrate analog circuits using electronic components.
3. To provide hands-on experience to the students so that they are able to put theoretical concepts to practice.
4. To use computer simulation tools such as PSPICE, or Multisim to the simulation of electronic circuits.
5. To create an ability to develop descriptions, explanations, predictions and models using evidence .
6. To create an ability to communicate effectively the scientific procedures and explanations about the experiments in oral/report forms.

List of Exercises/Experiments :

(Minimum 13 experiments are to be done in the semester, at least 6 each should be selected from the first(Exp. 1-10) and second(Exp. 11-20) half. Experiment no. 21 is compulsory).

1. Forward and reverse characteristics of PN diode and Zener diode
2. Input and output characteristics of BJT in CE configuration and evaluation of parameters
3. RC integrating and differentiating circuits-Transient response with different time constant
4. RC low pass and high pass circuits- Frequency response with sinusoidal input
5. Clipping circuits (Positive, negative and biased) - Transient and transfer characteristics
6. Clamping circuits (Positive, negative and biased)- Transient characteristics
7. Bridge Rectifier - with and without filter- ripple factor and regulation
8. Simple Zener regulator- Line and load characteristics
9. RC coupled CE amplifier – Mid band gain and frequency response
10. ~~Feedback amplifier (current series) - Mid band gain and frequency response~~
11. RC phase shift or Wien bridge oscillator using transistor
12. ~~Class B and class AB push pull power amplifier(Complementary symmetry)~~
13. Astable and Monostable multivibrators using transistors
14. Series voltage regulator (Two transistors)- Line and load characteristics
15. Voltage regulator using LM 723)- Line and load characteristics
16. Astable and monostable multivibrators using 555 Timer
17. Inverting and non-inverting amplifier using op-amp IC741
18. Instrumentation amplifier using op-amp IC741
19. ~~RC phase shift or Wien bridge oscillator using op-amp IC741~~
20. ~~Astable and Monostable multivibrators using op-amp IC741~~
21. Simulation of simple circuits (at least 6 from above) using any SPICE software(Transient, AC and DC analysis)

Course outcome:

Student is able to:

1. explain with basic electronic components and how to design and develop electronic circuits.
2. Design and demonstrate functioning of various discrete analog circuits
3. Familiar with computer simulation of electronic circuits and how to use it proficiently for design and development of electronic circuits.
4. Understand the concepts and their applications in engineering.
5. Communicate effectively the scientific procedures and explanations in formal technical presentations/reports.

SEMESTER - 4				
Course Code	***Course***	***L-T-P***	***Credits***	***Exam Slot***
~~MA206~~	~~Probability and Statistics~~	~~3-1-0~~	~~4~~	~~A~~
CS202	Computer Organization and Architecture	3-1-0	4	B
CS204	Operating Systems	3-1-0	4	C
CS206	Object Oriented Design and Programming	2-1-0	3	D
CS208	Principles of Database Design	2-1-0	3	E
CS232	Free and Open Source Software Lab	0-0-3	1	S
CS234	Digital System Lab	0-0-3	1	T
~~HS200~~	~~Business Economics~~	~~3-0-0~~	~~3~~	~~F~~
	Total Credits = 23 Hours 28/27			
	Cumulative Credits = 94			

Course No.	Course Name	L-T-P -Credits	Year of Introduction
CS 202	**Computer Organization and Architecture**	**3-1-0-4**	**2015**

Course Objectives

1. To impart an understanding of the internal organization and operations of a computer.
2. To introduce the concepts of processor logic design and control logic design.

Syllabus

Fundamental building blocks and functional units of a computer. Execution phases of an instruction. Arithmetic Algorithms. Design of the processing unit – how arithmetic and logic operations are performed. Design of the control unit – hardwired and microprogrammed control. I/O organisation – interrupts, DMA, different interface standards. Memory Subsystem – different types.

Expected outcome

At the end of the course,

1. Students would understand the basic structure and functioning of a digital computer.
2. The concepts of addressing and instruction execution cycle would enable the students to develop efficient programs.
3. Ability to design a basic processing unit using the concepts of ALU and control logic design.
4. Students could comment on the pros and cons of using a particular control logic design for a specific type of processor.
5. Students could choose the best suitable interfacing standard for a particular class of I/O device.
6. Ability to describe how the control logic, memory subsystem, I/O subsystem and processor functional units cooperate during instruction execution.

Text Books:

1. Hamacher C., Z. Vranesic and S. Zaky, *Computer Organization* ,5/e, McGraw Hill, 2011.

References:

- Mano M. M., Digital Logic & Computer Design, 4/e, Pearson Education, 2013.
- Patterson D.A. and J. L. Hennessey, Computer Organization and Design, 5/e, Morgan Kauffmann Publishers, 2013.
- Chaudhuri P., Computer Organization and Design, 2/e, Prentice Hall, 2008.
- Rajaraman V. and T. Radhakrishnan, Computer Organization and Architecture, Prentice Hall, 2011.
- Messmer H. P., The Indispensable PC Hardware Book, 4/e, Addison-Wesley, 2001

Course Plan

Module	Contents	Hours	Sem.ExamMarks

		(51)	
I	**Basic Structure of computers**–functional units –basic operational concepts –bus structures –software. Memory locations and addresses – memory operations – instructions and instruction sequencing – addressing modes – ARM Example (programs not required). Basic I/O operations – stacks subroutine calls.	6	15%
II	**Basic processing unit** – fundamental concepts – instruction cycle - execution of a complete instruction –multiple- bus organization – sequencing of control signals. **Arithmetic algorithms:** *Algorithms for addition and subtraction of binary and BCD numbers* — algorithms for multiplication and division of binary and BCD numbers — array multiplier —Booth's multiplication algorithm — restoring and non-restoring division — *algorithms for floating point addition, subtraction,* multiplication and division.	10	15%
FIRST INTERNAL EXAMINATION			
III	**Processor Logic Design:** Register transfer logic –inter register transfer – arithmetic, logic and shift micro operations –conditional control statements. **Processor organization:**–design of arithmetic unit, logic unit, arithmetic logic unit and shifter –status register –processor unit –design of accumulator.	9	15%
IV	**Control Logic Design:** Control organization –design of hardwired control –control of processor unit –PLA control. **Micro-programmed control:** Microinstructions –horizontal and vertical micro instructions – micro-program sequencer –micro programmed CPU organization.	9	15%
SECOND INTERNAL EXAMINATION			
V	**I/O organization:** accessing of I/O devices –interrupts –direct memory access –buses –interface circuits –standard I/O interfaces (PCI, SCSI, USB)	8	20%
VI	**Memory system :** basic concepts –semiconductor RAMs –memory system considerations –	9	20%

	semiconductor ROMs –flash memory –cache memory and mapping functions.		
END SEMESTER EXAM			

Course No.	Course Name	L-T-P -Credits	Year of Introduction
CS204	**Operating Systems**	**3-1-0-4**	**2015**

Course Objectives

1. To impart fundamental understanding of the purpose, structure, functions of operating system.
2. To impart the key design issues of an operating system

Syllabus

Basic concept of Operating System, its structure, Process management, inter-process communication, process synchronization, CPU Scheduling, deadlocks, Memory Management, swapping, segmentation, paging, Storage Management - disk scheduling, RAID, File System Interface-implementation. Protection.

Course outcome

At the end of the course, students shall be able to:

1. Explain how an operating system makes the life of a user much easier.
2. Explain and illustrate modern operating system calls such as Linux process creation and synchronization libraries.
3. Describe, contrast and compare several process and disk scheduling algorithms.
4. Explain, illustrate and analyze theory and implementation behind process, memory, file and I/O management.
5. Master several inter-process communication strategies.
6. Appreciate the need of access control in an operating system.

Text Book:

1. Abraham Silberschatz, Peter B Galvin, Gerg Gagne, Operating System Concepts, 9/e, Wiley India, 2015.

References:

1. Garry Nut, Operating Systems: 3/e, Pearson Education, 2004
2. Bhatt P. C. P., An Introduction to Operating Systems: Concepts and Practice, 3/e, Prentice Hall of India, 2010.
3. Madnick S. and J. Donovan, Operating Systems, McGraw Hill, 2001.
4. Hanson P. B., Operating System Principle, Prentice Hall of India, 2001.
5. Deitel H. M., An Introduction to Operating System Principles, Addison-Wesley, 1990.

Course Plan

Module	Contents	Hours (52)	Sem. ExamMarks

I	**Iintroduction**: What Operating Systems Do - Computer-System Organization - Computer-System Architecture - Operating-System Structure - Operating-System Operations – Overview. Kernel Data Structures - Computing Environments. **System Structures**: Operating-System Services - User and Operating-System Interface - System Calls – Examples of system calls used for various purpose Operating-System Design and Implementation - Operating-System Structure - System Boot.	10	15%
II	**Process Management**: Process Concept – Processes-States – Process Control Block – Threads. Scheduling – Queues – Schedulers – Context Switching. Operations – Creation and Termination. **Inter Process Communication**: Shared Memory, Message Passing, Sockets, Pipes, RPC. Threads – Overview.	10	15%
FIRST INTERNAL EXAMINATION			
III	**Process Synchronization**: Critical Section-Peterson's solution. Synchronization Hardware. Locks, Semaphores, Monitors, Classical Problems. **CPU Scheduling** – Scheduling Criteria – Scheduling Algorithms. **Deadlocks** – System Model, Characterization – Prevention – Avoidance – Detection- Recovery.	9	15%
IV	**Memory Management**: Main Memory – Background – Swapping – Contiguous Memory allocation – Segmentation – Paging – Structure of Page table.	8	15%
SECOND INTERNAL EXAMINATION			
V	**Storage Management:** *Overview of mass storage structure- disk structure – disk attachment.* Disk scheduling and management. Swap Space. RAID and different RAID levels (questions from this section should specify the function of RAID levels mentioned)	7	20%
VI	**File System Interface**: File Concepts – Attributes – operations – types – structure – access methods. File system mounting. Protection. File System structure – File system implementation. Directory implementation – allocation methods. Free space Management. **Protection**– Goals, Principles, Domain. Access Matrix	8	20%
END SEMESTER EXAM			

Course No.	Course Name	L-T-P - Credits	Year of Introduction
CS206	**Object Oriented Design and Programming**	**2-1-0-3**	**2015**

Course Objectives

1. To impart the basic concepts of object oriented design techniques.
2. To give a thorough understanding of Java language.
3. To provide basic exposure to the basics of multithreading, network programming, database connectivity etc.
4. To impart the techniques of creating GUI based applications.

Syllabus

Object oriented concepts, Object oriented systems development life cycle, Object oriented methodologies, Unified Modeling Language, Java Overview, Classes and objects, Parameter passing, Overloading, Inheritance, Overriding, Packages, Exception Handling, Input/Output, Threads and multithreading, Network programming using stream and datagram sockets, Applets, Event Handling mechanism, Working with frames and graphics, AWT Controls, Swings, Java database connectivity.

Course outcome.

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

1. Apply object oriented principles in software design process.
2. Develop Java programs for real applications using java constructs and libraries.
3. Develop multithreaded client/server applications using socket programming.
4. Create GUI based applications with database at back end.

Text Books:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Bahrami A., Object Oriented Systems Development using the Unified Modeling Language, McGraw Hill, 1999.

References:

1. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
2. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
3. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
4. Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.
6. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

Course Plan

Module	Contents	Hours (42)	Sem. ExamMarks
I	Object oriented concepts, Object oriented systems development life cycle. Object oriented methodologies: Rumbaugh methodology, Booch methodology, Jacobson et. al methodology. Unified Modeling Language, static and dynamic models, UML class diagram, Use-case diagram.	08	15%
II	Java Overview: Java virtual machine, *data types, operators, control statements,* introduction to Java programming. Classes fundamentals, objects, methods, constructors, parameter passing,	07	15%

	overloading, access control keywords.		
FIRST INTERNAL EXAMINATION			
III	Inheritance basics, method overriding, abstract classes, interfaces. Defining and importing packages. Exception handling fundamentals, multiple catch and nested try statements.	06	15%
IV	Input/Output: files, stream classes, reading console input. Threads: thread model, use of Thread class and Runnable interface, thread synchronization, multithreading.	06	15%
SECOND INTERNAL EXAMINATION			
V	Network programming: use of stream sockets and datagram sockets. Applet basics and methods. Event Handling: delegation event model, event classes, sources, listeners.	07	20%
VI	Introduction to AWT: working with frames, graphics, color, font. AWT Control fundamentals. Swing overview. Java database connectivity: JDBC overview, creating and executing queries, dynamic queries.	08	20%
END SEMESTER EXAM			

Course No.	Course Name	L-T-P -Credits	Year of Introduction
CS208	**Principles of Database Design**	**2-1-0-3**	**2015**

Course Objectives

1. To impart the basic understanding of the theory and applications of database management systems.
2. To give basic level understanding of internals of database systems.
3. To expose to some of the recent trends in databases.

Syllabus:

Types of data, database and DBMS, Languages and users. Software Architecture, E-R and Extended E-R Modelling, Relational Model – concepts and languages, relational algebra and tuple relational calculus, SQL, views, assertions and triggers, HLL interfaces, relational db design, FDs and normal forms, Secondary storage organization, indexing and hashing, query optimization, concurrent transaction processing and recovery principles, recent topics.

Course outcome.

At the end of the course, the student should be able to

1. define, explain and illustrate the fundamental concepts of databases.
2. model and design a relational database following the design principles.
3. develop queries for relational database.
4. Define, explain and illustrate fundamental principles of, data organization, query optimization and concurrent transaction processing.
5. list out and describe latest trends in databases.

Text Books:

1. Elmasri R. and S. Navathe, *Database Systems: Models, Languages, Design and Application Programming*, Pearson Education, 2013.
2. Sliberschatz A., H. F. Korth and S. Sudarshan, *Database System Concepts*, 6/e, McGraw Hill, 2011.

References:

1. Powers S., *Practical RDF*, O'Reilly Media, 2003.
2. Plunkett T., B. Macdonald, *et al.*, *Oracle Big Data Hand Book*, Oracle Press, 2013.

Course Plan

Module	Contents	Hours (42)	Sem. ExamMarks
I	**Introduction:** Data: structured, semi-structured and unstructured data**,** Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS. Database architectures and classification. (Reading: Elmasri Navathe, Ch. 1 and 2. Additional Reading:	**06**	**15%**

	Silbershatz, Korth, Ch. 1) **Entity-Relationship Model:** Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Relationships of degree greater than 2 (Reading:: Elmasri Navathe, Ch. 7.1-7.8)		
II	**Relational Model:** Structure of relational Databases, Integrity Constraints, synthesizing ER diagram to relational schema (Reading: Elmasri Navathe, Ch. 3 and 8.1, Additional Reading: Silbershatz, Korth, Ch. 2.1-2.4) **Database Languages:** Concept of DDL and DML relational algebra (Reading: Silbershatz, Korth, Ch 2.5-2.6 and 6.1-6.2, Elmasri Navathe, Ch. 6.1-6.5)	**06**	**15%**
FIRST INTERNAL EXAM			
III	**Structured Query Language (SQL)**: Basic SQL Structure, examples, Set operations, Aggregate Functions, nested sub-queries (Reading: Elmasri Navathe, Ch. 4 and 5.1) **Views, assertions and triggers** (Reading: Elmasri Navathe, Ch. 5.2-5.3, Silbershatz, Korth Ch. 5.3). **Functions, Procedures and HLL interfaces** (Reading: Silbershatz, Korth Ch. 5.1-5.2).	**07**	**15%**
IV	**Relational Database Design:** Different anomalies in designing a database, normalization, functional dependency, Armstrong's Axioms, closures, Equivalence of FDs, minimal Cover (proofs not required). Normalization using functional dependencies, INF, 2NF, 3NF and BCNF, lossless and dependency preserving decompositions (Reading: Elmasri and Navathe, Ch. 14.1-14.5, 15.1-15.2. Additional Reading: Silbershatz, Korth Ch. 8.1-8.5)	**07**	**15%**
SECOND INTERNAL EXAM			
V	**Physical Data Organization**: index structures, primary, secondary and clustering indices, hashing (Reading: Elmasri and Navathe, Ch. 16.1-16.7. 16.8.1, 16.8.2, 16.8.3 (extendible hashing only)). Single level and Multi-level indexing, B-Trees(basic structure only, algorithms not needed) and B+-Trees (basic structure only, algorithms not needed), Indexing on multiple keys (Reading Elmasri and Navathe, Ch. 17.1-17.4) **Query Optimization**: algorithms for relational algebra operations, heuristics-based query optimization, Cost-	**08**	**20%**

	based query optimization (Reading Elmasri and Navathe, Ch. 18.1-18.3, 18.6-18.8)		
VI	**Transaction Processing Concepts:** overview of concurrency control and recovery acid properties, serial and concurrent schedules, conflict serializability. Two-phase locking, failure classification, storage structure, stable storage, log based recovery, immediate and deferred database modification, check-pointing, (Reading Elmasri and Navathe, Ch. 20.1-20.5 (except 20.5.4-20.5.5) , Silbershatz, Korth Ch. 15.1 (except 15.1.4-15.1.5), Ch. 16.1 – 16.5) **Recent topics (preliminary ideas only):** Semantic Web and RDF(Reading: Powers Ch.1, 2), GIS, biological databases (Reading: Elmasri and Navathe Ch. 23.3-23.4) Big Data (Reading: Plunkett and Macdonald, Ch. 1, 2)	**08**	**20%**
END SEMESTER EXAM			

Course No.	Course Name	L-T-P-Credits	Year of Introduction
CS232	**Free and Open Source Software Lab**	**0-0-3-1**	**2015**

Course Objectives: To expose students to FOSS environment and introduce them to use open source
packages in open source platform.

List of Exercises/Experiments:

7. Getting started with Linux basic commands and directory structure, execute file, directory operations.
8. Linux commands for redirection, pipes, filters, job control, file ownership, file permissions, links and file system hierarchy.
9. Shell Programming : Write shell script to show various system configuration like
   - Currently logged user and his logname
   - Your current shell
   - Your home directory
   - Your operating system type
   - Your current path setting
   - Your current working directory
   - Show Currently logged number of users
10. Write shell script to show various system configuration like
    - About your os and version, release number, kernel version
    - Show all available shells
    - Show mouse settings
    - Show computer CPU information like processor type, speed etc
    - Show memory information
    - Show hard disk information like size of hard-disk, cache memory, model etc
    - File system (Mounted)
11. Shell script program for scientific calculator.
12. Write a script called addnames that is to be called as follows, where *classlist* is the name of the classlist file, and *username* is a particular student's username.

    ./addnames *classlist username*

    The script should

    4. check that the correct number of arguments was received and print an usage message if not,
    5. check whether the classlist file exists and print an error message if not,
    6. check whether the username is already in the file, and then either
    7. print a message stating that the name already existed, or
    8. add the name to the end of the list.

13. Version Control System setup and usage using GIT.
    - Creating a repository
    - Checking out a repository
    - Adding content to the repository
    - Committing the data to a repository
    - Updating the local copy
    - Comparing different revisions
    - Revert
    - Conflicts and Solving a conflict

14. Text processing and regular expression with Perl, Awk: simple programs, connecting with database e.g., MariaDB
15. Shell script to implement a script which kills every process which uses more than a specified value of memory or CPU and is run upon system start.
16. GUI programming : Create scientific calculator – using Gambas or try using GTK or QT
17. Running PHP : simple applications like login forms after setting up a LAMP stack
18. Advanced linux commands curl, wget, ftp, ssh and grep
19. Application deployment on a cloud-based LAMP stack/server with PHP eg: Openshift, Linode etc.
20. Kernel configuration, compilation and installation : Download / access the latest kernel source code from *kernel.org*, compile the kernel and install it in the local system.Try to view the source code of the kernel
21. Virtualisation environment (e.g., xen, kqemu, virtualbox or lguest) to test an applications, new kernels and isolate applications. It could also be used to expose students to other alternate OSs like *BSD
22. Compiling from source : learn about the various build systems used like the auto* family, cmake, ant etc. instead of just running the commands. This could involve the full process like fetching from a cvs and also include autoconf, automake etc.,
23. Introduction to packet management system : Given a set of RPM or DEB, how to build and maintain, serve packages over http or ftp. and also how do you configure client systems to access the package repository.
24. Installing various software packages. Either the package is yet to be installed or an older version is existing. The student can practice installing the latest version. Of course, this might need Internet access.
    - Install samba and share files to windows
    - Install Common Unix Printing System(CUPS)

Expected outcome:

Course No.	Course Name	L-T-P - Credits	Year of Introduction
CS234	**DIGITAL SYSTEM LAB**	**0-0-3-1**	**2015**

Course Objectives:

25. To familiarize students with digital ICs, the building blocks of digital circuits
26. To provide students the opportunity to set up different types of digital circuits and study their behaviour

List of Exercises/Experiments :

1. Familiarisation and verification of the truth tables of basic gates and universal gates.
2. Verification of Demorgan's laws for two variables.
3. Implementation of half adder and full adder circuits using logic gates.
4. Implementation of half subtractor and full subtractor circuits using logic gates.
5. Implementation of parallel adder circuit.
6. Realization of 4 bit adder/subtractor and BCD adder circuits using IC 7483.
7. Implementation of a 2 bit magnitude comparator circuit using logic gates.
8. Design and implementation of code convertor circuits
9. a) BCD to excess 3 code b) binary to gray code
10. Implementation of multiplexer and demultiplexer circuits using logic gates. Familiarization with various multiplexer and demultiplexer ICs.
11. Realization of combinational circuits using multiplexer/demultiplexer ICs.
12. Implementation of SR, D, JK, JK master slave and T flip flops using logic gates. Familiarization with IC 7474 and IC 7476.
13. Implementation of shift registers using flip flop Integrated Circuits.
14. Implementation of ring counter and Johnson counter using flip flop Integrated Circuits.
15. Realization of asynchronous counters using flip flop ICs.
16. Realization of synchronous counters using flip flop ICs. Familiarization with various counter Integrated Circuits.
17. Implementation of a BCD to 7 segment decoder and display.
18. Simulation of Half adder, Full adder using VHDL.

Course outcome:

After successful completion of this course students will

1. be familiar with the digital ICs and their use in implementing digital circuits.
2. gain practical experience in the design and implementation of different kinds of digital circuits.