

SCHEME OF TEACHING AND EXAMINATION 2015-2016

B.E. Computer Science & Engineering/ B.E. Information Science & Engineering

III SEMESTER

Sl. No	Subject Code	Title	Teaching Hours /Week		Examination				Credits
			Theory	Practical/ Drawing	Duration	Theory/ Practical Marks	I.A. Marks	Total Marks	
1	15MAT31	Engineering Mathematics - III	04	--	03	80	20	100	4
2	15CS32	Analog and Digital Electronics	04	--	03	80	20	100	4
3	15CS33	Data Structures and Applications	04	--	03	80	20	100	4
4	15CS34	Computer Organization	04	--	03	80	20	100	4
5	15CS35	Unix and Shell Programming	04	--	03	80	20	100	4
6	15CS36	Discrete Mathematical structures	04	--	03	80	20	100	4
7	15CSL37	Analog and Digital Electronics Laboratory	--	1I+2P	03	80	20	100	2
8	15CSL38	Data Structures Laboratory	--	1I+2P	03	80	20	100	2
TOTAL			24	6	24	640	160	800	28

Note : I Stands for Instruction Hours and P for practical Hours

Shah
H.O.D.
Dept. Of Information Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225

ENGINEERING MATHEMATICS-III

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2015 -2016)

SEMESTER – III

Subject Code	15MAT31	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Comprehend and use of analytical and numerical methods in different engineering fields
- Apprehend and apply Fourier Series
- Realize and use of Fourier transforms and Z-Transforms
- Use of statistical methods in curve fitting applications
- Use of numerical methods to solve algebraic and transcendental equations, vector integration and calculus of variation

Module -1	Teaching Hours
------------------	-----------------------

Fourier Series: Periodic functions, Dirichlet's condition, Fourier Series of Periodic functions with period 2π and with arbitrary period $2c$, Fourier series of even and odd functions, Half range Fourier Series, practical Harmonic analysis. Complex Fourier series

10Hours

Module -2

Fourier Transforms: Infinite Fourier transforms, Fourier Sine and Cosine transforms, Inverse transform. **Z-transform:** Difference equations, basic definition, z-transform - definition, Standard z-transforms, Damping rule, Shifting rule, Initial value and final value theorems (without proof) and problems, Inverse z-transform. Applications of z-transforms to solve difference equations.

10 Hours

Module – 3

Statistical Methods: Correlation and rank Correlation coefficients, Regression and Regression coefficients, lines of regression - problems **Curve fitting:** Curve fitting by the method of least squares, Fitting of the curves of the form, $y = ax + b$, $y = ax^2 + bx + c$, $y = ae^{bx}$, $y = ax^b$. **Numerical Methods:** Numerical solution of algebraic and transcendental equations by: Regular-falsi method, Secant method, Newton - Raphson method and Graphical method.

10 Hours

Module-4

Finite differences: Forward and backward differences, Newton's forward and backward interpolation formulae. Divided differences-Newton's divided difference formula. Lagrange's interpolation formula and inverse interpolation formula. Central Difference-Stirling's and Bessel's formulae (all formulae without proof)-Problems. **Numerical integration:** Simpson's 1/3, 3/8 rule, Weddle's rule (without proof) -Problems

10 Hours

Module-5	
<p>Vector integration: Line integrals-definition and problems, surface and volume integrals-definition, Green's theorem in a plane, Stokes and Gauss-divergence theorem (without proof) and problems.</p> <p>Calculus of Variations: Variation of function and Functional, variational problems, Euler's equation, Geodesics, minimal surface of revolution, hanging chain, problems</p>	10 Hours
Course outcomes:	
<p>After Studying this course, students will be able to</p> <ul style="list-style-type: none"> • Use of periodic signals and Fourier series to analyze circuits • Explain the general linear system theory for continuous-time signals and systems using the Fourier Transform • Analyze discrete-time systems using convolution and the z-transform • Use appropriate numerical methods to solve algebraic and transcendental equations and also to calculate a definite integral • Use curl and divergence of a vector function in three dimensions, as well as apply the Green's Theorem, Divergence Theorem and Stokes' theorem in various applications • Solve the simple problem of the calculus of variations 	
Graduate Attributes (as per NBA)	
<ol style="list-style-type: none"> 1. Engineering Knowledge 2. Problem Analysis 3. Life-Long Learning 4. Conduct Investigations of Complex Problems 	
Question paper pattern:	
<p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>	
Text Books:	
<ol style="list-style-type: none"> 1. B. S. Grewal, "Higher Engineering Mathematics", Khanna publishers, 42nd edition, 2013. 2. B.V. Ramana "Higher Engineering Mathematics" Tata McGraw-Hill, 2006. 	
Reference Books:	
<ol style="list-style-type: none"> 1. N. P. Bali and Manish Goyal, "A text book of Engineering mathematics", Laxmi publications, latest edition. 2. Kreyszig, "Advanced Engineering Mathematics" - 9th edition, Wiley. 3. H. K Dass and Er. Rajnish Verma, "Higher Engineering Mathematics", S. Chand, 1st ed. 	

Sashy
H.O.D.

ANALOG AND DIGITAL ELECTRONICS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2015 -2016)

SEMESTER - III

Subject Code	15CS32	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable the students to

- Recall and Recognize construction and characteristics of JFETs and MOSFETs and differentiate with BJT
- Evolve and Analyze Operational Amplifier circuits and their applications
- Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques.
- Describe and Design Decoders, Encoders, Digital multiplexers, Adders and Subtractors, Binary comparators, Latches and Master-Slave Flip-Flops.
- Describe, Design and Analyze Synchronous and Asynchronous Sequential
- Explain and design registers and Counters, A/D and D/A converters.

Module -1

Teaching Hours

Field Effect Transistors: Junction Field Effect Transistors, MOSFETs, Differences between JFETs and MOSFETs, Biasing MOSFETs, FET Applications, CMOS Devices. Wave-Shaping Circuits: Integrated Circuit(IC) Multivibrators. **Introduction to Operational Amplifier:** Ideal v/s practical Opamp, Performance Parameters, **Operational Amplifier Application Circuits:** Peak Detector Circuit, Comparator, Active Filters, Non-Linear Amplifier, Relaxation Oscillator, Current-To-Voltage Converter, Voltage-To-Current Converter.
Text book 1:- Ch5: 5.2, 5.3, 5.5, 5.8, 5.9, 5.1.Ch13: 13.10.Ch 16: 16.3, 16.4. Ch 17: 7.12, 17.14, 17.15, 17.18, 17.19, 17.20, 17.21.)

10 Hours

Module -2

The Basic Gates: Review of Basic Logic gates, Positive and Negative Logic, Introduction to HDL. **Combinational Logic Circuits:** Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Hazards and Hazard covers, HDL Implementation Models.
Text book 2:- Ch2: 2.4, 2.5. Ch3: 3.2 to 3.11.

10 Hours

Module – 3

Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit Flip- Flops: RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIP-FLOPs. Text book 2:- Ch 4:- 4.1 to 4.9, 4.11, 4.12, 4.14.Ch6:-6.7, 6.10.Ch8:- 8.1 to 8.5.	10 Hours
Module-4	
Flip- Flops: FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, HDL Implementation of FLIP-FLOP. Registers: Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register implementation in HDL. Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus. (Text book 2:- Ch 8: 8.6, 8.8, 8.9, 8.10, 8.13. Ch 9: 9.1 to 9.8. Ch 10: 10.1 to 10.4)	10 Hours
Module-5	
Counters: Decade Counters, Presetable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL. D/A Conversion and A/D Conversion: Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, Continuous A/D Conversion, A/D Techniques, Dual-slope A/D Conversion, A/D Accuracy and Resolution. Text book 2:- Ch 10: 10.5 to 10.9. Ch 12: 12.1 to 12.10	10 Hours
Course outcomes: After Studying this course, students will be able to	
<ul style="list-style-type: none"> • Explain the operation of JFETs and MOSFETs , Operational Amplifier circuits and their application • Explain Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique. • Demonstrate Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors, working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters • Design of Counters, Registers and A/D & D/A converters 	
Graduate Attributes (as per NBA) <ol style="list-style-type: none"> 1. Engineering Knowledge 2. Design/Development of Solutions(partly) 3. Modern Tool Usage 4. Problem Analysis 	
Question paper pattern: The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books:	

1. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, 2012.
2. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015

Reference Books:

1. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design with VHDL, 2nd Edition, Tata McGraw Hill, 2005.
2. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine-Pearson, 2010.
3. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson, 2008.



H. O. D.

Dept. Of Information Science & Engineering
Aiva's Institute of Engg. & Technology
Nijar, MOOBBIDRI - 574 225

DATA STRUCTURES AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)

SEMESTER - III

Subject Code	15CS33	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS - 04

Course objectives: This course will enable students to

- Explain fundamentals of data structures and their applications essential for programming/problem solving
- Analyze Linear Data Structures: Stack, Queues, Lists
- Analyze Non-Linear Data Structures: Trees, Graphs
- Analyze and Evaluate the sorting & searching algorithms
- Assess appropriate data structure during program development/Problem Solving

Module -1

**Teaching
Hours**

Introduction: Data Structures, Classifications (Primitive & Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, Dynamically allocated arrays, **Array Operations:** Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials and Sparse Matrices. **Strings:** Basic Terminology, Storing, Operations and Pattern Matching algorithms. Programming Examples.

Text 1: Ch 1: 1.2, Ch 2: 2.2 -2.7

Text 2: Ch 1: 1.1 -1.4, Ch 3: 3.1-3.3,3.5,3.7, Ch 4: 4.1-4.9,4.14

Ref 3: Ch 1: 1.4

10 Hours

Module -2

Stacks and Queues

Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression, **Recursion** - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues:** Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. Multiple Stacks and Queues. Programming Examples.

Text 1: Ch 3: 3.1 -3.7

Text 2: Ch 6: 6.1 -6.3, 6.5, 6.7-6.10, 6.12, 6.13

10 Hours

Module - 3

<p>Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of - Linked lists – Polynomials, Sparse matrix representation. Programming Examples Text 1: Ch 4: 4.1 –4.8 except 4.6 Text 2: Ch 5: 5.1 – 5.10</p>	<p>10 Hours</p>
<p>Module-4</p>	
<p>Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples Text 1: Ch 5: 5.1 –5.5, 5.7 Text 2: Ch 7: 7.1 – 7.9</p>	<p>10 Hours</p>
<p>Module-5</p>	
<p>Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. Sorting and Searching: Insertion Sort, Radix sort, Address Calculation Sort. Hashing: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. Files and Their Organization: Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing Text 1: Ch 6: 6.1 –6.2, Ch 7:7.2, Ch 8:8.1-8.3 Text 2: Ch 8: 8.1 – 8.7, Ch 9:9.1-9.3,9.7,9.9 Reference 2: Ch 16: 16.1 - 16.7</p>	<p>10 Hours</p>
<p>Course outcomes:</p>	
<p>After studying this course, students will be able to:</p> <ul style="list-style-type: none"> • Acquire knowledge of <ul style="list-style-type: none"> - Various types of data structures, operations and algorithms. - Sorting and searching operations. - File structures. • Analyse the performance of <ul style="list-style-type: none"> - Stack, Queue, Lists, Trees, Graphs, Searching and Sorting techniques. • Implement all the applications of Data structures in a high-level language. • Design and apply appropriate data structures for solving computing problems. 	
<p>Graduate Attributes (as per NBA)</p>	
<ol style="list-style-type: none"> 1. Engineering Knowledge 2. Design/Development of Solutions 3. Conduct Investigations of Complex Problems 4. Problem Analysis 	

Question paper pattern:

The question paper will have ten questions.

There will be 2 questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Fundamentals of Data Structures in C - Ellis Horowitz and Sartaj Sahni, 2nd edition, Universities Press, 2014
2. Data Structures - Seymour Lipschutz, Schaum's Outlines, Revised 1st edition, McGraw Hill, 2014

Reference Books:

1. Data Structures: A Pseudo-code approach with C -Gilberg & Forouzan, 2nd edition, Cengage Learning, 2014.
2. Data Structures using C, , Reema Thareja, 3rd edition Oxford press, 2012.
3. An Introduction to Data Structures with Applications- Jean-Paul Tremblay & Paul G. Sorenson, 2nd Edition, McGraw Hill, 2013.
4. Data Structures using C - A M Tenenbaum, PHI, 1989.
5. Data Structures and Program Design in C - Robert Kruse, 2nd edition, PHI, 1996.

Stahly
H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225

COMPUTER ORGANIZATION [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2015 -2016) SEMESTER - III			
Subject Code	15CS34	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Understand the basics of computer organization: structure and operation of computers and their peripherals. • Understand the concepts of programs as sequences or machine instructions. • Expose different ways of communicating with I/O devices and standard I/O interfaces. • Describe hierarchical memory systems including cache memories and virtual memory. • Describe arithmetic and logical operations with integer and floating-point operands. • Understand basic processing unit and organization of simple processor, concept of pipelining and other large computing systems. 			
Module -1			Teaching Hours
Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions Textbook 1: Ch 1: 1.3, 1.4, 1.6.1, 1.6.2, 1.6.4, 1.6.7. Ch 2: 2.2 to 2.10, 2.12			10Hours
Module -2			
Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB. Textbook 1: Ch 4: 4.1, 4.2: 4.2.1 to 4.2.5, 4.4 to 4.7.			10 Hours
Module – 3			
Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories, Secondary Storage. Textbook 1: Ch 5: 5.1 to 5.4, 5.5.1, 5.5.2, 5.6, 5.7, 5.9			10 Hours
Module-4			

<p>Arithmetic: Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division, Floating-point Numbers and Operations.</p> <p>Textbook 1: Ch 2: 2.1, Ch 6: 6.1 to 6.7</p>	<p>10 Hours</p>
<p>Module-5</p>	
<p>Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Micro programmed Control. Embedded Systems and Large Computer Systems: Examples of Embedded Systems, Processor chips for embedded applications, Simple Microcontroller. The structure of General-Purpose Multiprocessors.</p> <p>Textbook 1: Ch 7: 7.1 to 7.5, Ch 9:9.1 to 9.3, Ch 12:12.3</p>	<p>10 Hours</p>
<p>Course outcomes:</p>	
<p>After studying this course, students will be able to:</p> <ul style="list-style-type: none"> • Acquire knowledge of <ul style="list-style-type: none"> - The basic structure of computers & machine instructions and programs, Addressing Modes, Assembly Language, Stacks, Queues and Subroutines. - Input/output Organization such as accessing I/O Devices, Interrupts. - Memory system basic Concepts, Semiconductor RAM Memories, Static memories, Asynchronous DRAMS, Read Only Memories, Cache Memories and Virtual Memories. - Some Fundamental Concepts of Basic Processing Unit, Execution of a Complete Instruction, Multiple Bus Organization, Hardwired Control and Micro programmed Control. - Pipelining, embedded and large computing system architecture. • Analyse and design arithmetic and logical units. • Apply the knowledge gained in the design of Computer. • Design and evaluate performance of memory systems • Understand the importance of life-long learning 	
<p>Graduate Attributes (as per NBA)</p> <ol style="list-style-type: none"> 1. Engineering Knowledge 2. Problem Analysis 3. Life-Long Learning 	
<p>Question paper pattern:</p> <p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>	
<p>Text Books:</p> <ol style="list-style-type: none"> 1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th Edition, Tata McGraw Hill, 2002. 	
<p>Reference Books:</p> <ol style="list-style-type: none"> 1. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015. 	

Signature
H. O. D.

UNIX AND SHELL PROGRAMMING [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2015 -2016) SEMESTER – III			
Subject Code	15CS35	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Understand the UNIX Architecture, File systems and use of basic Commands. • Use of editors and Networking commands. • Understand Shell Programming and to write shell scripts. • Understand and analyze UNIX System calls, Process Creation, Control & Relationship. 			
Module -1			Teaching Hours
Introduction, Brief history. Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. The login prompt. General features of Unix commands/ command structure. Command arguments and options. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The man command knowing more about Unix commands and using Unix online manual pages. The man with keyword option and whatis. The more command and using it with other commands. Knowing the user terminal, displaying its characteristics and setting characteristics. Managing the non-uniform behaviour of terminals and keyboards. The root login. Becoming the super user: su command. The /etc/passwd and /etc/shadow files. Commands to add, modify and delete users.			10Hours
Topics from chapter 2 , 3 and 15 of text book 1,chapter 1 from text book 2			
Module -2			
Unix files. Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. File attributes and permissions and knowing them. The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions.			10Hours
Topics from chapters 4, 5 and 6 of text book 1			

Module – 3	
<p>The vi editor. Basics. The .exrc file. Different ways of invoking and quitting vi. Different modes of vi. Input mode commands. Command mode commands. The ex mode commands. Illustrative examples Navigation commands. Repeat command. Pattern searching. The search and replace command. The set, map and abbr commands. Simple examples using these commands.</p> <p>The shells interpretive cycle. Wild cards and file name generation. Removing the special meanings of wild cards. Three standard files and redirection. Connecting commands: Pipe. Splitting the output: tee. Command substitution. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions.</p> <p>Topics from chapters 7, 8 and 13 of text book 1. Topics from chapter 2 and 9 ,10 of text book 2</p>	10Hours
Module-4	
<p>Shell programming. Ordinary and environment variables. The .profile. Read and readonly commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples. File inodes and the inode structure. File links – hard and soft links. Filters. Head and tail commands. Cut and paste commands. The sort command and its usage with different options. The umask and default file permissions. Two special files /dev/null and /dev/tty.</p> <p>Topics from chapter 11, 12, 14 of text book 1,chapter 17 from text book2</p>	10Hours
Module-5	
<p>Meaning of a process. Mechanism of process creation. Parent and child process. The ps command with its options. Executing a command at a specified point of time: at command. Executing a command periodically: cron command and the crontab file.. Signals. The nice and nohup commands. Background processes. The bg and fg command. The kill command. The find command with illustrative example.</p> <p>Structure of a perl script. Running a perl script. Variables and operators. String handling functions. Default variables - \$_ and \$. – representing the current line and current line number. The range operator. Chop() and chomp() functions. Lists and arrays. The @-variable. The splice operator, push(), pop(), split() and join(). File handles and handling file – using open(), close() and die () functions.. Associative arrays – keys and value functions. Overview of decision making loop control structures – the foreach. Regular expressions – simple and multiple search patterns. The match and substitute operators. Defining and using subroutines.</p> <p>Topics from chapter 9 and 19 of text book 1. Topics from chapter 11 of reference book 1</p>	10Hours

Course outcomes:
After studying this course, students will be able to: <ul style="list-style-type: none">• Explain multi user OS UNIX and its basic features• Interpret UNIX Commands, Shell basics, and shell environments• Design and develop shell programming, communication, System calls and terminology.• Design and develop UNIX File I/O and UNIX Processes.• Perl script writing
Graduate Attributes (as per NBA) <ol style="list-style-type: none">1. Engineering Knowledge2. Environment and Sustainability3. Design/Development of Solutions
Question paper pattern: <p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>
Text Books: <ol style="list-style-type: none">1. Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill2. Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009.
Reference Books: <ol style="list-style-type: none">1. M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2nd Edition , Wiley, 2014.

Sreehitha
H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225

DISCRETE MATHEMATICAL STRUCTURES [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2015 -2016) SEMESTER – III			
Subject Code	15CS36	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Prepare for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science. • Understand and apply logic, relations, functions, basic set theory, countability and counting arguments, proof techniques, • Understand and apply mathematical induction, combinatorics, discrete probability, recursion, sequence and recurrence, elementary number theory • Understand and apply graph theory and mathematical proof techniques. 			
Module -1			Teaching Hours
Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems, Textbook 1: Ch 2			10Hours
Module -2			
Properties of the Integers: Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions. Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition, Textbook 1: Ch 4: 4.1, 4.2 Ch 1.			10 Hours
Module – 3			
Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. Textbook 1: Ch 5:5.1 to 5.3, 5.5, 5.6, Ch 7:7.1 to 7.4			10 Hours
Module-4			

<p>The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials. Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients.</p> <p>Textbook 1: Ch 8: 8.1 to 8.4, Ch 10:10.1 to 10.2</p>	<p>10 Hours</p>
<p>Module-5</p>	
<p>Introduction to Graph Theory: Definitions and Examples, Sub graphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits , Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes</p> <p>Textbook 1: Ch 11: 11.1 to 11.3, Ch 12: 12.1 to 12.4</p>	<p>10 Hours</p>
<p>Course outcomes:</p>	
<p>After studying this course, students will be able to:</p> <ol style="list-style-type: none"> 1. Verify the correctness of an argument using propositional and predicate logic and truth tables. 2. Demonstrate the ability to solve problems using counting techniques and combinatorics in the context of discrete probability. 3. Solve problems involving recurrence relations and generating functions. 4. Construct proofs using direct proof, proof by contraposition, proof by contradiction, proof by cases, and mathematical induction. 5. Explain and differentiate graphs and trees 	
<p>Graduate Attributes (as per NBA)</p> <ol style="list-style-type: none"> 1. Engineering Knowledge 2. Problem Analysis 3. Conduct Investigations of Complex Problems 	
<p>Question paper pattern:</p> <p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>	
<p>Text Books:</p> <ol style="list-style-type: none"> 1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, , 5th Edition, Pearson Education. 2004. 	
<p>Reference Books:</p> <ol style="list-style-type: none"> 1. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics – A Concept based approach, Universities Press, 2016 2. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007. 3. Jayant Ganguly: A Treatise on Discrete Mathematical Structures, Sanguine-Pearson, 2010. 4. D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004. 5. Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008. 	

Shahid
H.O.D.

ANALOG AND DIGITAL ELECTRONICS LABORATORY [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2015 -2016) SEMESTER - III			
Laboratory Code	15CSL37	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03
CREDITS – 02			
<p>Course objectives: This laboratory course enable students to get practical experience in design, assembly and evaluation/testing of</p> <ul style="list-style-type: none"> • Analog components and circuits including Operational Amplifier, Timer, etc. • Combinational logic circuits. • Flip - Flops and their operations • Counters and Registers using Flip-flops. • Synchronous and Asynchronous Sequential Circuits. • A/D and D/A Converters 			
<p>Descriptions (if any)</p> <p><i>Any simulation package like MultiSim / P-spice /Equivalent software may be used.</i></p> <p>Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.</p> <p>Laboratory Session-1: Write-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.</p> <p>Laboratory Session-2: Write-upon Logic design components, pin diagram (if any), Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.</p> <p>Note: <i>These TWO Laboratory sessions</i> are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 20 marks as lab experiments.</p>			

Laboratory Experiments:

1. a) Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.
b) Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.
2. a) Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.
b) Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.
3. Design and implement an Astable multivibrator circuit using 555 timer for a given frequency and duty cycle.

NOTE: hardware and software results need to be compared

Continued:

4. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.
5. a) Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
b) Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.
6. a) Design and implement code converter I) Binary to Gray (II) Gray to Binary Code using basic gates.
7. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.
8. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
b) Design and develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.
9. a) Design and implement a mod-n ($n < 8$) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.
10. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ($n \leq 9$) and demonstrate on 7-segment display (using IC-7447).
11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).

Study experiment

12. To study 4-bit ALU using IC-74181.

Course outcomes:

On the completion of this laboratory course, the students will be able to:


- Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- Design and demonstrate various combinational logic circuits.
- Design and demonstrate various types of counters and Registers using Flip-flops
- Use simulation package to design circuits.
- Understand the working and implementation of ALU.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (1 to 11 nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:
 - a) For questions having part a only- Procedure + Conduction + Viva: 20 + 50 + 10 = 80 Marks
 - b) For questions having part a and b
 - Part a- Procedure + Conduction + Viva: 10 + 35 + 05 = 50 Marks
 - Part b- Procedure + Conduction + Viva: 10 + 15 + 05 = 30 Marks
5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.


 Dept. Of Information Science & Engineering
 Alva's Institute of Engg. & Technology
 Mijar, MOODBIDRI - 574 225

DATA STRUCTURES LABORATORY [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2015 -2016) SEMESTER - III			
Laboratory Code	15CSL38	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03
CREDITS - 02			
Course objectives: This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of <ul style="list-style-type: none"> • Asymptotic performance of algorithms. • Linear data structures and their applications such as Stacks, Queues and Lists • Non-Linear Data Structures and their Applications such as Trees and Graphs • Sorting and Searching Algorithms 			
Descriptions (if any) <p style="text-align: center;">Implement all the experiments in C Language under Linux / Windows environment.</p>			
Laboratory Experiments: <ol style="list-style-type: none"> Design, Develop and Implement a menu driven Program in C for the following Array operations <ol style="list-style-type: none"> Creating an Array of N Integer Elements Display of Array Elements with Suitable Headings Inserting an Element (ELEM) at a given valid Position (POS) Deleting an Element at a given valid Position(POS) Exit. <p>Support the program with functions for each of the above operations.</p> Design, Develop and Implement a Program in C for the following operations on Strings <ol style="list-style-type: none"> Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR <p>Support the program with functions for each of the above operations. Don't use Built-in functions.</p> Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) <ol style="list-style-type: none"> Push an Element on to Stack Pop an Element from Stack Demonstrate how Stack can be used to check Palindrome Demonstrate Overflow and Underflow situations on Stack 			

- e. Display the status of Stack
- f. Exit

Support the program with appropriate functions for each of the above operations

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, %(Remainder), ^(Power) and alphanumeric operands.
5. Design, Develop and Implement a Program in C for the following Stack Applications
 - a. Evaluation of **Suffix expression** with single digit operands and operators: +, -, *, /, %, ^
 - b. Solving **Tower of Hanoi** problem with **n** disks
6. Design, Develop and Implement a menu driven Program in C for the following operations on **Circular QUEUE** of Characters (Array Implementation of Queue with maximum size **MAX**)
 - a. Insert an Element on to Circular QUEUE
 - b. Delete an Element from Circular QUEUE
 - c. Demonstrate **Overflow** and **Underflow** situations on Circular QUEUE
 - d. Display the status of Circular QUEUE
 - e. Exit

Support the program with appropriate functions for each of the above operations

Continued:

7. Design, Develop and Implement a menu driven Program in C for the following operations on **Singly Linked List (SLL)** of Student Data with the fields: **USN, Name, Branch, Sem, PhNo**
 - a. Create a **SLL** of **N** Students Data by using **front insertion**.
 - b. Display the status of **SLL** and count the number of nodes in it
 - c. Perform Insertion / Deletion at End of **SLL**
 - d. Perform Insertion / Deletion at Front of **SLL**(**Demonstration of stack**)
 - e. Exit
8. Design, Develop and Implement a menu driven Program in C for the following operations on **Doubly Linked List (DLL)** of Employee Data with the fields: **SSN, Name, Dept, Designation, Sal, PhNo**
 - a. Create a **DLL** of **N** Employees Data by using **end insertion**.
 - b. Display the status of **DLL** and count the number of nodes in it
 - c. Perform Insertion and Deletion at End of **DLL**
 - d. Perform Insertion and Deletion at Front of **DLL**
 - e. Demonstrate how this **DLL** can be used as **Double Ended Queue**
 - f. Exit

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

CHOICE BASED CREDIT SYSTEM (CBCS)


SCHEME OF TEACHING AND EXAMINATION 2015-2016

B.E. Computer Science & Engineering/ B.E. Information Science & Engineering

IV SEMESTER

Sl. No	Subject Code	Title	Teaching Hours /Week		Examination				Credits
			Theory	Practical/ Drawing	Duration	Theory/ Practical Marks	I.A. Marks	Total Marks	
1	15MAT41	Engineering Mathematics - IV	04	--	03	80	20	100	4
2	15CS 42	Software Engineering	04	--	03	80	20	100	4
3	15CS43	Design and Analysis of Algorithms	04	--	03	80	20	100	4
4	15CS 44	Microprocessors and Microcontrollers	04	--	03	80	20	100	4
5	15CS45	Object Oriented Concepts	04	--	03	80	20	100	4
6	15CS46	Data Communication	04	--	03	80	20	100	4
7	15CSL47	Design and Analysis of Algorithm Laboratory	--	1I+2P	03	80	20	100	2
8	15CSL48	Microprocessors Laboratory	--	1I+2P	03	80	20	100	2
TOTAL			24	06	24	640	160	800	28

Note: 'I' Stands for Instruction Hours and 'P' for practical Hours


H.O.D.
 Dept. Of Information Science & Engineering
 Alva's Institute of Engg. & Technology
 Mijar, MOODBIDRI - 574 225

ENGINEERING MATHEMATICS-IV [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15MAT41	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Formulate, solve and analyze engineering problems. • Apply numerical methods to solve ordinary differential equations. • Apply finite difference method to solve partial differential equations. • Perform complex analysis. • Interpret use of sampling theory. • Apply joint probability distribution and stochastic process. 			
Module 1			Teaching Hours
Numerical Methods: Numerical solution of ordinary differential equations of first order and first degree, Picard's method, Taylor's series method, modified Euler's method, Runge-Kutta method of fourth order. Milne's and Adams-Bashforth predictor and corrector methods (No derivations of formulae). Numerical solution of simultaneous first order ordinary differential equations, Picard's method, Runge-Kutta method of fourth order			10 Hours
Module 2			
Numerical Methods: Numerical solution of second order ordinary differential equations, Picard's method, Runge-Kutta method and Milne's method. Special Functions: Bessel's functions- basic properties, recurrence relations, orthogonality and generating functions. Legendre's functions - Legendre's polynomial, Rodrigue's formula, problems.			10 Hours
Module 3			
Complex Variables: Function of a complex variable, limits, continuity, differentiability, Analytic functions-Cauchy-Riemann equations in Cartesian and polar forms. Properties and construction of analytic functions. Complex line integrals-Cauchy's theorem and Cauchy's integral formula, Residue, poles, Cauchy's Residue theorem with proof and problems. Transformations: Conformal transformations, discussion of transformations: $w = z^2$, $w = e^z$, $w = z + (a^2/z)$ and bilinear transformations.			10 Hours
Module 4			
Probability Distributions: Random variables (discrete and continuous), probability functions. Poisson distributions, geometric distribution, uniform distribution, exponential and normal distributions, Problems. Joint probability distribution: Joint Probability distribution for two variables, expectation, covariance, correlation coefficient.			10 Hours
Module 5			
Sampling Theory: Sampling, Sampling distributions, standard error, test of hypothesis for means and proportions, confidence limits for means, student's t-distribution, Chi-square distribution as a test of goodness of fit. Stochastic process: Stochastic process, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains, higher transition probability.			10 Hours

<p>Course Outcomes: After studying this course, students will be able to:</p> <ul style="list-style-type: none"> • Use appropriate numerical methods to solve first and second order ordinary differential equations. • Use Bessel's and Legendre's function which often arises when a problem possesses axial and spherical symmetry, such as in quantum mechanics, electromagnetic theory, hydrodynamics and heat conduction. • State and prove Cauchy's theorem and its consequences including Cauchy's integral formula. • Compute residues and apply the residue theorem to evaluate integrals. • Analyze, interpret, and evaluate scientific hypotheses and theories using rigorous statistical methods.
<p>Graduate Attributes</p> <ul style="list-style-type: none"> • Engineering Knowledge • Problem Analysis • Life-Long Learning • Conduct Investigations of Complex Problems
<p>Question paper pattern:</p> <p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>
<p>Text Books:</p> <ol style="list-style-type: none"> 1. B.V.Ramana "Higher Engineering Mathematics" Tata McGraw-Hill, 2006. 2. B. S. Grewal, "Higher Engineering Mathematics", Khanna publishers, 42nd edition, 2013.
<p>Reference Books:</p> <ol style="list-style-type: none"> 1. N P Bali and Manish Goyal, "A text book of Engineering mathematics", Laxmi publications, latest edition. 2. Kreyszig, "Advanced Engineering Mathematics" - 9th edition, Wiley, 2013. 3. H. K Dass and Er. Rajnish Verma, "Higher Engineering Mathematics", S. Chand, 1st ed, 2011.

Sice hell
H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225

SOFTWARE ENGINEERING [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15CS42	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to			
<ul style="list-style-type: none"> • Outline software engineering principles and activities involved in building large software programs. • Identify ethical and professional issues and explain why they are of concern to software engineers. • Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation. • Differentiate system models, use UML diagrams and apply design patterns. • Discuss the distinctions between validation testing and defect testing. • Recognize the importance of software maintenance and describe the intricacies involved in software evolution. • Apply estimation techniques, schedule project activities and compute pricing. • Identify software quality parameters and quantify software using measurements and metrics. • List software quality standards and outline the practices involved. • Recognize the need for agile software development, describe agile methods, apply agile practices and plan for agility. 			
Module 1			Teaching Hours
Introduction: Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics. Case Studies. Software Processes: Models: Waterfall Model (Sec 2.1.1), Incremental Model (Sec 2.1.2) and Spiral Model (Sec 2.1.3). Process activities. Requirements Engineering: Requirements Engineering Processes (Chap 4). Requirements Elicitation and Analysis (Sec 4.5). Functional and non-functional requirements (Sec 4.1). The software Requirements Document (Sec 4.2). Requirements Specification (Sec 4.3). Requirements validation (Sec 4.6). Requirements Management (Sec 4.7).			12 Hours
Module 2			
System Models: Context models (Sec 5.1). Interaction models (Sec 5.2). Structural models (Sec 5.3). Behavioral models (Sec 5.4). Model-driven engineering (Sec 5.5). Design and Implementation: Introduction to RUP (Sec 2.4), Design Principles (Chap 17). Object-oriented design using the UML (Sec 7.1). Design patterns (Sec 7.2). Implementation issues (Sec 7.3). Open source development (Sec 7.4).			11 Hours
Module 3			
Software Testing: Development testing (Sec 8.1), Test-driven development (Sec 8.2), Release testing (Sec 8.3), User testing (Sec 8.4). Test Automation (Page no 42, 70,212, 231,444,695). Software Evolution: Evolution processes (Sec 9.1). Program evolution dynamics (Sec			9 Hours

9.2). Software maintenance (Sec 9.3). Legacy system management (Sec 9.4).	
Module 4	
Project Planning: Software pricing (Sec 23.1). Plan-driven development (Sec 23.2). Project scheduling (Sec 23.3): Estimation techniques (Sec 23.5). Quality management: Software quality (Sec 24.1). Reviews and inspections (Sec 24.3). Software measurement and metrics (Sec 24.4). Software standards (Sec 24.2)	10 Hours
Module 5	
Agile Software Development: Coping with Change (Sec 2.3), The Agile Manifesto: Values and Principles. Agile methods: SCRUM (Ref "The SCRUM Primer, Ver 2.0") and Extreme Programming (Sec 3.3). Plan-driven and agile development (Sec 3.2). Agile project management (Sec 3.4), Scaling agile methods (Sec 3.5):	8 Hours
Course Outcomes: After studying this course, students will be able to:	
<ul style="list-style-type: none"> • Design a software system, component, or process to meet desired needs within realistic constraints. • Assess professional and ethical responsibility • Function on multi-disciplinary teams • Use the techniques, skills, and modern engineering tools necessary for engineering practice • Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems. 	
Graduate Attributes	
<ul style="list-style-type: none"> • Project Management and Finance • Conduct Investigations of Complex Problems • Modern Tool Usage • Ethics 	
Question paper pattern:	
<p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>	
Text Books:	
<ol style="list-style-type: none"> 1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24) 2. The SCRUM Primer, Ver 2.0, http://www.goodagile.com/scrumpriemer/scrumpriemer20.pdf 	
Reference Books:	
<ol style="list-style-type: none"> 1. Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill. 2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India 	
Web Reference for eBooks on Agile:	
<ol style="list-style-type: none"> 1. http://agilemanifesto.org/ 2. http://www.jamesshore.com/Agile-Book/ 	

Sce hdd

H.O.D.

Dept. Of Information Science & Engineering
 Aiva's Institute of Engg. & Technology
 Mijar, MOODBIDRI - 574 225

Downloaded from <http://gnanadeevige.com>

DESIGN AND ANALYSIS OF ALGORITHMS [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15CS43	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to			
<ul style="list-style-type: none"> • Explain various computational problem solving techniques. • Apply appropriate method to solve a given problem. • Describe various methods of algorithm analysis. 			
Module 1			Teaching Hours
Introduction: What is an Algorithm? (T2:1.1), Algorithm Specification (T2:1.2), Analysis Framework (T1:2.1), Performance Analysis: Space complexity, Time complexity (T2:1.3). Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), and Little-oh notation (o), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples (T1:2.2, 2.3, 2.4). Important Problem Types: Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. Fundamental Data Structures: Stacks, Queues, Graphs, Trees, Sets and Dictionaries. (T1:1.3,1.4)			10 Hours
Module 2			
Divide and Conquer: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum (T2:3.1, 3.3, 3.4), Merge sort, Quick sort (T1:4.1, 4.2), Strassen's matrix multiplication (T2:3.8), Advantages and Disadvantages of divide and conquer. Decrease and Conquer Approach: Topological Sort. (T1:5.3)			10 Hours
Module 3			
Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines (T2:4.1, 4.3, 4.5). Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm (T1:9.1, 9.2). Single source shortest paths: Dijkstra's Algorithm (T1:9.3). Optimal Tree problem: Huffman Trees and Codes (T1:9.4). Transform and Conquer Approach: Heaps and Heap Sort (T1:6.4).			10 Hours
Module 4			
Dynamic Programming: General method with Examples, Multistage Graphs (T2:5.1, 5.2). Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem ((T1:8.2, 8.3, 8.4), Bellman-Ford Algorithm (T2:5.4), Travelling Sales Person problem (T2:5.9), Reliability design (T2:5.8).			10 Hours
Module 5			
Backtracking: General method (T2:7.1), N-Queens problem (T1:12.1), Sum of subsets problem (T1:12.1), Graph coloring (T2:7.4), Hamiltonian cycles (T2:7.5). Branch and Bound: Assignment Problem, Travelling Sales Person problem (T1:12.2), 0/1 Knapsack problem (T2:8.2, T1:12.2): LC Branch and Bound solution (T2:8.2), FIFO Branch and Bound solution (T2:8.2). NP-Complete and NP-Hard problems: Basic			10 Hours

concepts, non-deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes (T2:11.1).	
Course Outcomes: After studying this course, students will be able to	
<ul style="list-style-type: none"> • Describe computational solution to well known problems like searching, sorting etc. • Estimate the computational complexity of different algorithms. • Devise an algorithm using appropriate design strategies for problem solving. 	
Graduate Attributes	
<ul style="list-style-type: none"> • Engineering Knowledge • Problem Analysis • Design/Development of Solutions • Conduct Investigations of Complex Problems • Life-Long Learning 	
Question paper pattern:	
<p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>	
Text Books:	
<p>T1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009. Pearson. T2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press</p>	
Reference Books:	
<ol style="list-style-type: none"> 1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI 2. Design and Analysis of Algorithms , S. Sridhar, Oxford (Higher Education) 	

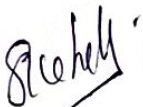
Sachin

H.O.D.

Dept. Of Information Science & Engineering
Aiva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225

MICROPROCESSORS AND MICROCONTROLLERS [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15CS44	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Make familiar with importance and applications of microprocessors and microcontrollers • Expose architecture of 8086 microprocessor and ARM processor • Familiarize instruction set of ARM processor 			
Module 1			Teaching Hours
The x86 microprocessor: Brief history of the x86 family, Inside the 8088/86, Introduction to assembly programming, Introduction to Program Segments, The Stack, Flag register, x86 Addressing Modes. Assembly language programming: Directives & a Sample Program, Assemble, Link & Run a program, More Sample programs, Control Transfer Instructions, Data Types and Data Definition, Full Segment Definition, Flowcharts and Pseudo code. Text book 1: Ch 1: 1.1 to 1.7, Ch 2: 2.1 to 2.7			10 Hours
Module 2			
x86: Instructions sets description, Arithmetic and logic instructions and programs: Unsigned Addition and Subtraction, Unsigned Multiplication and Division, Logic Instructions, BCD and ASCII conversion, Rotate Instructions. INT 21H and INT 10H Programming : Bios INT 10H Programming , DOS Interrupt 21H. 8088/86 Interrupts, x86 PC and Interrupt Assignment. Text book 1: Ch 3: 3.1 to 3.5, Ch 4: 4.1 , 4.2 Chapter 14: 14.1 and 14.2			10 Hours
Module 3			
Signed Numbers and Strings: Signed number Arithmetic Operations, String operations. Memory and Memory interfacing: Memory address decoding, data integrity in RAM and ROM, 16-bit memory interfacing. 8255 I/O programming: I/O addresses MAP of x86 PC's, programming and interfacing the 8255. Text book 1: Ch 6: 6.1, 6.2. Ch 10: 10.2, 10.4, 10.5. Ch 11: 11.1 to 11.4			10 Hours
Module 4			
Microprocessors versus Microcontrollers, ARM Embedded Systems : The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software, ARM Processor Fundamentals : Registers , Current Program Status Register , Pipeline, Exceptions, Interrupts, and the Vector Table , Core Extensions Text book 2: Ch 1:1.1 to 1.4, Ch 2:2.1 to 2.5			10 Hours
Module 5			
Introduction to the ARM Instruction Set : Data Processing Instructions , Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants, Simple programming exercises. Text book 2: Ch 3:3.1 to 3.6 (Excluding 3.5.2)			10 Hours
Course Outcomes: After studying this course, students will be able to			

<ul style="list-style-type: none"> • Differentiate between microprocessors and microcontrollers • Design and develop assembly language code to solve problems • Gain the knowledge for interfacing various devices to x86 family and ARM processor • Demonstrate design of interrupt routines for interfacing devices
Graduate Attributes
<ul style="list-style-type: none"> • Engineering Knowledge • Problem Analysis • Design/Development of Solutions
Question paper pattern:
<p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>
Text Books:
<ol style="list-style-type: none"> 1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Danny Causey, The x86 PC Assembly Language Design and Interfacing, 5th Edition, Pearson, 2013. 2. ARM system developers guide, Andrew N Sloss, Dominic Symes and Chris Wright, Elsevier, Morgan Kaufman publishers, 2008.
Reference Books:
<ol style="list-style-type: none"> 1. Douglas V. Hall: Microprocessors and Interfacing, Revised 2nd Edition, TMH, 2006. 2. K. Udaya Kumar & B.S. Umashankar : Advanced Microprocessors & IBM-PC Assembly Language Programming, TMH 2003. 3. Ayala : The 8086 Microprocessor: programming and interfacing - 1st edition, Cengage Learning 4. The Definitive Guide to the ARM Cortex-M3, by Joseph Yiu, 2nd Edition , Newnes, 2009 5. The Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd., 1st edition, 2005 6. ARM System-on-Chip Architecture, Steve Furber, Second Edition, Pearson, 2015 7. Architecture, Programming and Interfacing of Low power Processors- ARM7, Cortex-M and MSP430, Lyla B Das Cengage Learning, 1st Edition


H.O.D.
 Dept. Of Information Science & Engineering
 Aiva's Institute of Engineering & Technology
 Mijar, MOOBBIDRI - 574 225

OBJECT ORIENTED CONCEPTS [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15CS45	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Learn fundamental features of object oriented language and JAVA • Set up Java JDK environment to create, debug and run simple Java programs. • Create multi-threaded programs and event handling mechanisms. • Introduce event driven Graphical User Interface (GUI) programming using applets and swings. 			
Module 1			Teaching Hours
Introduction to Object Oriented Concepts: A Review of structures, Procedure–Oriented Programming system, Object Oriented Programming System, Comparison of Object Oriented Language with C, Console I/O, variables and reference variables, Function Prototyping, Function Overloading. Class and Objects: Introduction, member functions and data, objects and functions, objects and arrays, Namespaces, Nested classes, Constructors, Destructors. Text book 1: Ch 1: 1.1 to 1.9 Ch 2: 2.1 to 2.6 Ch 4: 4.1 to 4.2			10 Hours
Module 2			Teaching Hours
Introduction to Java: Java's magic: the Byte code; Java Development Kit (JDK); the Java Buzzwords, Object-oriented programming; Simple Java programs. Data types, variables and arrays, Operators, Control Statements. Text book 2: Ch:1 Ch: 2 Ch:3 Ch:4 Ch:5			10 Hours
Module 3			Teaching Hours
Classes, Inheritance, Exceptions, Packages and Interfaces: Classes: Classes fundamentals; Declaring objects; Constructors, this keyword, garbage collection. Inheritance: inheritance basics, using super, creating multi level hierarchy, method overriding. Exception handling: Exception handling in Java. Packages, Access Protection, Importing Packages, Interfaces. Text book 2: Ch:6 Ch: 8 Ch:9 Ch:10			10 Hours
Module 4			Teaching Hours
Multi Threaded Programming, Event Handling: Multi Threaded Programming: What are threads? How to make the classes threadable ; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, read-write problem, producer consumer problems. Event Handling: Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes. Text book 2: Ch 11: Ch: 22			10 Hours
Module 5			Teaching Hours
The Applet Class: Introduction, Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting;			10 Hours

Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase(); AppletContext and showDocument(); The AudioClip Interface; The AppletStub Interface; Output to the Console. Swings: Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; JLabel and ImageIcon; JTextField; The Swing Buttons; JTabbedPane; JScrollPane; JList; JComboBox; JTable. Text book 2: Ch 21: Ch: 29 Ch: 30	
Course Outcomes: After studying this course, students will be able to	
<ul style="list-style-type: none"> • Explain the object-oriented concepts and JAVA. • Develop computer programs to solve real world problems in Java. • Develop simple GUI interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles using Applets and swings. 	
Graduate Attributes	
<ul style="list-style-type: none"> • Programming Knowledge • Design/Development of Solutions • Conduct Investigations of Complex Problems • Life-Long Learning 	
Question paper pattern:	
<p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>	
Text Books:	
<ol style="list-style-type: none"> 1. Sourav Sahay, Object Oriented Programming with C++ , Oxford University Press, 2006 (Chapters 1, 2, 4) 2. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007. (Chapters 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 21, 22, 29, 30) 	
Reference Book:	
<ol style="list-style-type: none"> 1. Mahesh Bhavne and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008, ISBN: 9788131720806 2. Herbert Schildt, The Complete Reference C++, 4th Edition, Tata McGraw Hill, 2003. 3. Stanley B. Lippmann, Josee Lajore, C++ Primer, 4th Edition, Pearson Education, 2005. 4. Rajkumar Buyya, S Thamaras Selvi, Xingchen Chu, Object oriented Programming with Java, Tata McGraw Hill Education Private Limited. 5. Richard A Johnson, Introduction to Java Programming and OOAD, CENGAGE Learning. 6. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies. 	
Note: Every institute shall organize a bridge organize on C++ either in the vacation or in the beginning of even semester.	


H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engg & Technology
Mijar, MOODBIDRI - 574 225

DATA COMMUNICATION [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15CS46	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
CREDITS – 04			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Comprehend the transmission technique of digital data between two or more computers and a computer network that allows computers to exchange data. • Explain with the basics of data communication and various types of computer networks; • Illustrate TCP/IP protocol suite and switching criteria. • Demonstrate Medium Access Control protocols for reliable and noisy channels. • Expose wireless and wired LANs along with IP version. 			
Contents			Teaching Hours
Module 1			
Introduction: Data Communications, Networks, Network Types, Internet History, Standards and Administration, Networks Models: Protocol Layering, TCP/IP Protocol suite, The OSI model, Introduction to Physical Layer-1: Data and Signals, Digital Signals, Transmission Impairment, Data Rate limits, Performance, Digital Transmission: Digital to digital conversion (Only Line coding: Polar, Bipolar and Manchester coding).			10 Hours
Module 2			
Physical Layer-2: Analog to digital conversion (only PCM), Transmission Modes, Analog Transmission: Digital to analog conversion, Bandwidth Utilization: Multiplexing and Spread Spectrum, Switching: Introduction, Circuit Switched Networks and Packet switching.			10 Hours
Module 3			
Error Detection and Correction: Introduction, Block coding, Cyclic codes, Checksum, Forward error correction, Data link control: DLC services, Data link layer protocols, HDLC, and Point to Point protocol (Framing, Transition phases only).			10 Hours
Module 4			
Media Access control: Random Access, Controlled Access and Channelization, Wired LANs Ethernet: Ethernet Protocol, Standard Ethernet, Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet, Wireless LANs: Introduction, IEEE 802.11 Project and Bluetooth.			10 Hours
Module 5			
Other wireless Networks: WIMAX, Cellular Telephony, Satellite networks, Network layer Protocols : Internet Protocol, ICMPv4, Mobile IP, Next generation IP: IPv6 addressing, The IPv6 Protocol, The ICMPv6 Protocol and Transition from IPv4 to IPv6.			10 Hours
Course Outcomes: After studying this course, students will be able to <ul style="list-style-type: none"> • Illustrate basic computer network technology. • Identify the different types of network topologies and protocols. • Enumerate the layers of the OSI model and TCP/IP functions of each layer. • Make out the different types of network devices and their functions within a network 			

<ul style="list-style-type: none"> Demonstrate the skills of subnetting and routing mechanisms.
Graduate Attributes
<ol style="list-style-type: none"> Engineering Knowledge Design Development of solution(Partly) Modern Tool Usage Problem Analysis
Question paper pattern:
<p>The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</p>
Text Book:
<p>Behrouz A. Forouzan, Data Communications and Networking 5E, 5th Edition, Tata McGraw-Hill, 2013. (Chapters 1.1 to 1.5, 2.1 to 2.3, 3.1, 3.3 to 3.6, 4.1 to 4.3, 5.1, 6.1, 6.2, 8.1 to 8.3, 10.1 to 10.5, 11.1 to 11.4, 12.1 to 12.3, 13.1 to 13.5, 15.1 to 15.3, 16.1 to 16.3, 19.1 to 19.3, 22.1 to 22.4)</p>
Reference Books:
<ol style="list-style-type: none"> Alberto Leon-Garcia and Indra Widjaja: Communication Networks - Fundamental Concepts and Key architectures, 2nd Edition Tata McGraw-Hill, 2004. William Stallings: Data and Computer Communication, 8th Edition, Pearson Education, 2007. Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2007. Nader F. Mir: Computer and Communication Networks, Pearson Education, 2007

Signature

H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225

DESIGN AND ANALYSIS OF ALGORITHM LABORATORY [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – IV			
Subject Code	15CSL47	IA Marks	20
Number of Lecture Hours/Week	01 I + 02 P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03
CREDITS – 02			
Course objectives: This course will enable students to <ul style="list-style-type: none"> • Design and implement various algorithms in JAVA • Employ various design strategies for problem solving. • Measure and compare the performance of different algorithms. 			
Description Design, develop, and implement the specified algorithms for the following problems using Java language under LINUX /Windows environment.Netbeans/Eclipse IDE tool can be used for development and demonstration.			
Experiments			
1	A	Create a Java class called <i>Student</i> with the following details as variables within it. (i) USN (ii) Name (iii) Branch (iv) Phone Write a Java program to create <i>nStudent</i> objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.	
	B	Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working.	
2	A	Design a superclass called <i>Staff</i> with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely <i>Teaching</i> (domain, publications), <i>Technical</i> (skills), and <i>Contract</i> (period). Write a Java program to read and display at least 3 <i>staff</i> objects of all three categories.	
	B	Write a Java class called <i>Customer</i> to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as <name, dd/mm/yyyy> and display as <name, dd, mm, yyyy> using StringTokenizer class considering the delimiter character as "/".	
3	A	Write a Java program to read two integers <i>a</i> and <i>b</i> . Compute <i>a/b</i> and print, when <i>b</i> is not zero. Raise an exception when <i>b</i> is equal to zero.	
	B	Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number.	
4		Sort a given set of <i>n</i> integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of <i>n</i> > 5000 and record the time taken to sort. Plot a graph of the time taken versus <i>n</i> on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.	

5	Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of $n > 5000$, and record the time taken to sort. Plot a graph of the time taken versus n on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.
6	Implement in Java, the 0/1 Knapsack problem using (a) Dynamic Programming method (b) Greedy method.
7	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm . Write the program in Java.
8	Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm . Use Union-Find algorithms in your program.
9	Find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm .
10	Write Java programs to (a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm . (b) Implement Travelling Sales Person problem using Dynamic programming.
11	Design and implement in Java to find a subset of a given set $S = \{S_1, S_2, \dots, S_n\}$ of n positive integers whose SUM is equal to a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$, there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. Display a suitable message, if the given problem instance doesn't have a solution.
12	Design and implement in Java to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

Course Outcomes: The students should be able to:

- Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
- Implement a variety of algorithms such as sorting, graph related, combinatorial, etc., in a high level language.
- Analyze and compare the performance of algorithms using language features.
- Apply and implement learned algorithm design techniques and data structures to solve real-world problems.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Modern Tool Usage
- Conduct Investigations of Complex Problems
- Design/Development of Solutions

Conduction of Practical Examination:

All laboratory experiments (Twelve problems) are to be included for practical examination. Students are allowed to pick one experiment from the lot.

To generate the data set use random number generator function.

Strictly follow the instructions as printed on the cover page of answer script for breakup of marks

Marks distribution: Procedure + Conduction + Viva: 20 + 50 + 10 (80). Change of experiment is allowed only once and marks allotted to the procedure

Signature
H.O.D.

Dept. Of Information Science & Engg.
Alva's Institute of Engineering & Tech.
MIDC, POBIDRI - 574 222

MICROPROCESSOR AND MICROCONTROLLER LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

SEMESTER – IV

Subject Code	15CSL48	IA Marks	20
Number of Lecture Hours/Week	01 I + 02 P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- To provide practical exposure to the students on microprocessors, design and coding knowledge on 80x86 family/ARM. To give the knowledge and practical exposure on connectivity and execute of interfacing devices with 8086/ARM kit like LED displays, Keyboards, DAC/ADC, and various other devices.

Description

Demonstration and Explanation hardware components and Faculty in-charge should explain 8086 architecture, pin diagram in one slot. The second slot, the Faculty in-charge should explain instruction set types/category etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-up on Microprocessors, 8086 Functional block diagram, Pin diagram and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-up on Instruction group, Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are evaluated as lab experiments for 20 marks.

Experiments

- Develop and execute the following programs using 8086 Assembly Language. Any suitable assembler like MASM/TASM/8086 kit or any equivalent software may be used.
- Program should have suitable comments.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.
- Software Required: Open source ARM Development platform, KEIL IDE and Proteus for simulation

SOFTWARE PROGRAMS: PART A

- Design and develop an assembly language program to search a key element "X" in a list of 'n' 16-bit numbers. Adopt Binary search algorithm in your program for searching.
- Design and develop an assembly program to sort a given set of 'n' 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.
- Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.
- Develop an assembly language program to compute nCr using recursive procedure. Assume that 'n' and 'r' are non-negative integers.

5. Design and develop an assembly language program to read the current time and Date from the system and display it in the standard format on the screen.
6. To write and simulate ARM assembly language programs for data transfer, arithmetic and logical operations (Demonstrate with the help of a suitable program).
7. To write and simulate C Programs for ARM microprocessor using KEIL (Demonstrate with the help of a suitable program)

Note : To use KEIL one may refer the book: **Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005**

HARDWARE PROGRAMS: PART B

8. a. Design and develop an assembly program to demonstrate BCD Up-Down Counter (00-99) on the Logic Controller Interface.
b. Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display $X*Y$.
9. Design and develop an assembly program to display messages "FIRE" and "HELP" alternately with flickering effects on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor is it necessary for the student to compute these values).
10. Design and develop an assembly program to drive a Stepper Motor interface and rotate the motor in specified direction (clockwise or counter-clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).
11. Design and develop an assembly language program to
 - a. Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).
 - b. Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).
12. To interface LCD with ARM processor-- ARM7TDMI/LPC2148. Write and execute programs in C language for displaying text messages and numbers on LCD
13. To interface Stepper motor with ARM processor-- ARM7TDMI/LPC2148. Write a program to rotate stepper motor

Study Experiments:

1. Interfacing of temperature sensor with ARM freedom board (or any other ARM microprocessor board) and display temperature on LCD
2. To design ARM cortex based automatic number plate recognition system
3. To design ARM based power saving system

Course Outcomes: After studying this course, students will be able to

- Learn 80x86 instruction sets and gains the knowledge of how assembly language works.
- Design and implement programs written in 80x86 assembly language
- Know functioning of hardware devices and interfacing them to x86 family
- Choose processors for various kinds of applications.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Modern Tool Usage
- Conduct Investigations of Complex Problems
- Design/Development of Solutions

Conduction of Practical Examination:

- All laboratory experiments (all 7 + 6 nos) are to be included for practical examination.
- Students are allowed to pick one experiment from each of the lot.
- Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
- PART -A: Procedure + Conduction + Viva: 10 + 25 +05 (40)
- PART -B: Procedure + Conduction + Viva: 10 + 25 +05 (40)
- Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

Steele
H.O.D.

Dept. Of Information Science & Engineering
Alva's Institute of Engineering & Technology
Mijar, MOODEBIDRI - 574 225