Academic Program: UG

Academic Year 2019-20

Syllabus

III & IV Semester B.E.

Information Science and Engineering

SHRI DHARMASTHALA MANJUNATHESHWARA COLLEGE OF ENGINEERING & TECHNOLOGY,

DHARWAD – 580 002

(An Autonomous Institution recognized by AICTE & Affiliated to VTU, Belagavi)

Ph: **0836-2447465 Fax: 0836-2464638 Web:** www.sdmcet.ac.in

SDM College of Engineering & Technology, Dharwad

Date: 18-7-2019

It is certified that the scheme and syllabus for **III & IV** semester of UG program in **Information Science & Engineering** is recommended by Board of Studies of **Information Science & Engineering** Department and approved by the Academic Council, SDM College of Engineering &Technology, Dharwad. This scheme and syllabus will be in force from the academic year 2019-20 till further revision.

Principal Chairman BoS & HoD

SDM College of Engineering & Technology, Dharwad
Department of **Information Science & Engineering**
(Our motto: Innovation through Information Technology)

College Vision and Mission

Vision:
To develop competent professionals with human values.

Mission:
1. To have contextually relevant Curricula.
2. To promote effective Teaching Learning Practices supported by Modern Educational Tools and Techniques.
3. To enhance Research Culture.
4. To involve Industrial Expertise for connecting classroom content to real life situations.
5. To inculcate Ethics and impart soft-skills leading to overall Personality Development.

SDMCET- Quality Policy
- In its quest to be a role model institution, committed to meet or exceed the utmost interest of all the stake holders.

SDMCET- Core Values
- Competency
- Commitment
- Equity
- Team work and
- Trust

Department Vision and Mission

Vision:

- To develop competent Engineers with Knowledge and skills in contemporary Information Technology practices.

Mission:

- To develop contemporary curriculum in information technology delivered using Innovative teaching learning practices and ICT tools.
- To promote research and expose students and faculty to best industry Practices.
- To inculcate moral values and professional ethics in students.

Programme Educational Objectives (PEOs):

- Develop into Information Technology Professionals with expertise in providing solutions to Information Engineering problems.
- Pursue higher studies with the sound knowledge of basic concepts and skills in basic science, humanities and Information Technology disciplines.
- Exhibit professionalism, ethics and ability to work in teams.

Program Outcomes (POs):

Sl. No	(A) Description of Program Outcomes
Engineering Graduates will demonstrate:	
1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	**Design/development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
(B) Description of Program Specific Outcomes (PSOs)	
13	An ability to develop logical reasoning, coding skills, analysis and mathematical modeling.
14	An ability to modify, debug, test and adapt software modules for varied applications.

Scheme for III Semester

Course Code	Course Category	Course Title	Teaching		Examination				
			L-T-P (Hrs/Week)	Credits	CIE	Theory (SEE)		Practical (SEE)	
					Max. Marks	*Max. Marks	Duration in hours	Max. Marks	Duration in hours
18UMAC300	BS	Engineering Mathematics-III	3 - 0 - 0	3	50	100	3	-	-
18UISC300	PC	Data Structures	3 - 2 - 0	4	50	100	3	-	-
18UISC301	PC	Logic Design	3 - 0 - 0	3	50	100	3	-	-
18UISC302	PC	Discrete Mathematics & Graph Theory	4 - 0 - 0	4	50	100	3	-	-
18UISC303	PC	Unix and Shell Programming	3 - 0 - 2	4	50	100	3	-	-
18UISC304	PC	Computer Organization and Architecture	3 - 0 - 0	3	50	100	3	-	-
18UISL305	PC	Data Structures Laboratory	0 - 0 - 3	1.5	50	-	-	50	3
18UISL306	PC	Logic Design Laboratory	0 - 0 -3	1.5	50	-	-	50	3
Total			**19 - 2 - 8**	**24**	**400**	**600**		**100**	

CIE: Continuous Internal Evaluation **SEE**: Semester End Examination*

L: Lecture **T**: Tutorials **P**: Practical

*SEE for theory courses is conducted for 100 marks and reduced to 50 marks.

BS- Basic Science, PC- Program Core

Scheme for IV Semester

Course Code	Course Category	Course Title	Teaching		Examination				
			L-T-P (Hrs/Week)	Credits	CIE	Theory (SEE)		Practical (SEE)	
					Max. Marks	*Max. Marks	Duration in hours	Max. Marks	Duration In hours
18UMAC400	BS	Engineering Mathematics - IV	3 - 0 - 0	3	50	100	3	-	-
18UISC400	PC	Object Oriented Programming	4 - 0 - 0	4	50	100	3	-	-
18UISC401	PC	Microcontroller	4 - 0 - 0	4	50	100	3	-	-
18UISC402	PC	Finite Automata and Formal Language	3 - 2 - 0	4	50	100	3	-	-
18UISC403	PC	Design and Analysis of Algorithms	3 - 0 - 0	3	50	100	3	-	-
18UISC404	PC	Operating System	3 - 0 - 0	3	50	100	3	-	-
18UISL405	PC	Object Oriented Programming Laboratory	0 - 0 - 3	1.5	50	-	-	50	3
18UISL406	PC	Microcontroller Laboratory	0 - 0 -3	1.5	50	-	-	50	3
18UISL407	PC	Introductory Project	0 - 0-2	1	50	-	-	-	-
Total			**20 - 2 -8**	**25**	**450**	**600**		**100**	

CIE: Continuous Internal Evaluation **SEE**: Semester End Examination* **L**: Lecture **T**: Tutorials **P**: Practical

*SEE for theory courses is conducted for 100 marks and reduced to 50 marks.

BS- Basic Science, PC- Program Core

Total Credits offered for the Second year: 49

III Semester

18UMAC300 Engineering Mathematics III (3-0-0) 3

Contact Hours: 39

Course Learning Objectives (CLOs):

To have an insight into Laplace transforms, Fourier series, Fourier transforms, Difference equations and Z-transforms. To develop the proficiency in variational calculus and solving ODE's arising in engineering applications, using numerical methods.

Course Outcomes (COs):

COs	**Description of the course outcomes:** At the end of course the students will be able to:	**Mapping to POs (1-12)**		
		Mastering 3	**Moderate 2**	**Introductory 1**
CO-1	Transform the given function using Laplace /Fourier transforms depending on the nature of engineering applications.			1
CO-2	Express periodic function as a Fourier series and obtain the various harmonics of the Fourier series expansion for the given numerical data.			1,2
CO-3	Solve difference equations using Z-transform.			1
CO-4	Solve first and second order ordinary differential equations arising in engineering problems using single step and multistep numerical methods.		1,2	
CO-5	Determine the externals of functional using calculus of variations and solve problems arising in engineering.			1,2

POs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mapping Level	1.2	1.3													

Contents:

1) Laplace Transforms: Definition and Properties. Laplace transform of elementary functions. Laplace transforms of Periodic functions and unit-step function – problems.

Inverse Laplace Transforms: Inverse Laplace transform - problems, Convolution theorem(without proof) to find the inverse Laplace transform and problems, solution of linear differential equations using Laplace transform. **8 Hrs.**

2) Fourier Series: Periodic functions, Dirichlet's condition. Fourier series of periodic functions of period 2π and arbitrary period. Half- range Fourier series. Practical harmonic analysis, examples from engineering field. **8 Hrs.**

3) Fourier Transforms: Infinite Fourier transforms, Fourier sine and cosine transforms. Inverse Fourier transforms. Simple problems.

Z-Transforms and Difference Equations : Z-transform- definition, Standard Z-transforms, Damping and shifting rules, Initial value and Final value theorems (without proof) with problems. Inverse Z-transform. Simple problems. Difference equations-basic definition. Application of Z-transform to solve Difference equation. **8 Hrs.**

4) Numerical Solutions of Ordinary Differential Equations (ODE's): Numerical solution of ODE's of first order and first degree- Taylor's series method, Modified Euler's method. Runge –Kutta method of fourth order, Milne's predictor and corrector method (No derivations of formulae). Problems. **7 Hrs.**

5) Numerical Solution of Second Order ODE's: Runge-Kutta method and Milne's predictor and Corrector method. (No derivations of formulae). Calculus of Variations: Variation of function and functional, variational problems, Euler's equation (without proof), Geodesics(plane), hanging chain problems. **8 Hrs.**

Text Books:

1) B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 44th Ed., 2017.

2) E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed. (Reprint).2016.

3) Srimanta Pal et al: Engineering Mathematics, Oxford University Press, 3rd Edition, 2016.

Reference Books:

1) C. Ray Wylie, Louis C. Barrett : "Advanced Engineering Mathematics", 6th Edition, McGraw-Hill Book Co., New York, 1995.

2) S. S. Sastry, Introductory Methods of Numerical Analysis, Prentice Hall of India, 4th Edition 2010.

3) B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.

4) N. P. Bali and Manish Goyal : A Text Book of Engineering Mathematics, Laxmi Publishers, 7th Ed., 2014.

5) Veerarajan T., "Engineering Mathematics for First year", Tata McGraw-Hill, 2008.

6) Thomas G.B. and Finney R.L."Calculus and Analytical Geometry" 9th Edition, Pearson, 2012.

Web links and Video Lectures:

http://nptel.ac.in/courses.php?disciplineID=111.
http://www.class-central.com/subject/math(MOOCs).
http://academicearth.org/.
VTU EDUSAT PROGRAMME – 20.

18UISC300	Data Structures	(3-2-0) 4

Contact Hours: 52

Course Learning Objectives (CLOs): The objective of the course is to realize the fundamental data structures like stacks, queues, linked list, trees. Also to compare and contrast the costs and benefits of dynamic and static data structure implementations. Students should be able to select an appropriate data structure for modeling a given problem.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Explain and illustrate usage of arrays, pointers, structures and unions.	1		
CO-2	Explain and illustrate usage of various linear data structures.	1		
CO-3	Explain and illustrate usage of various non-linear data structure.	1		
CO-4	Write modular programs for problem solving using either recursive or non-recursive functions.	13		14
CO-5	Write programs using appropriate linear data structures for various applications.	13	12	14
CO-6	Write programs using appropriate	13	12	14

	nonlinear data structures for various applications.													
PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS 13	PSO14
Mapping Level	3.0											2	3.0	1.0

Prerequisites:

1. Background of C programming

Contents:

1. **Introduction:** Elementary Data Organizations, Arrays, Functions, Pointers, structures and unions. Searching: Linear Search and Binary Search Techniques. **9L+4T=13Hrs**
2. **Stacks and Queues:** ADT Stack and its operations: Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues. **10L+3T=13Hrs**
3. **Linked Lists:** Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it. Circular Linked Lists **10L+3T=13Hrs**
4. **Trees**: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, Tree operations on each of the trees and their applications. **10L+3T=13Hrs**

Reference books:

[1] Y. Langsam, M. J. Augenstein and A. M. Tenenbaum, “Data Structures using C and C++”, 2/e, Pearson Education, 2015.

[2] Richard F. Gilberg, Behrouz A. Forouzan, “Data Structures – A Pseudocode Approach with C”, 2/e, Thomson India, 2005.

[3] AnandaRao, Radhika Raju Palagiri,”Data Structrures and Algorithms using C++”,Pearson Education, 2010.

[4] “Fundamentals of Data Structures”, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.

18UISC301 Logic Design (3-0-0) 3

Contact Hours: 39

Course Learning Objectives (CLOs): This course will enable students to:

- Make use of simplifying techniques in the design of combinational circuits.

- Illustrate combinational and sequential digital circuits
- Demonstrate the use of flip-flops and apply for registers
- Design and test counters, Peripheral Devices like ADC, DAC, Actuators and Sensors

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Illustrate** the basic elements of logic circuits	1		
CO-2	**Interface** various peripheral devices with logic circuits	1		3
CO-3	**Analyze** the problem and choose appropriate technique for solving the problem		2	
CO-4	**Design** various combinational and sequential circuits for a given problem	3		12

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3.0	1.0	2.0									1.0		

Prerequisites: Number systems and fundamentals of digital electronics.

Contents:

1. **Combinational Logic Circuits:** Boolean laws and Theorems**,** 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. **8 Hrs.**
2. **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, **7 Hrs.**
3. **Flip- Flops:** RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge- triggered D FLIP-FLOPs, Edge-triggered JK FLIP-FLOPs, FLIP-FLOP Timing, JK Master- slave FLIP-FLOP, HDL Implementation of FLIP-FLOP. **6 Hrs.**
4. **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. **6 Hrs.**
5. **Counters:** Asynchronous Counters, Decoding Gates, Synchronous Counters,

Changing the Counter Modulus, Decade Counters. **6 Hrs.**

6. **Peripheral Devices:** ADC, DAC, Actuators and Sensors. **6 Hrs.**

Reference Books:

[1] Donald P Leach, Albert Paul Malvino & Goutam Saha, Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015.

[2] M. Morris Mano, Digital Design, 4th Edition, Pearson Prentice Hall, 2008.

[3] Sensors and Actuators: Engineering System Instrumentation, Second Edition 2nd Edition by Clarence W. de Silva (Author)

18UISC302 Discrete Mathematics & Graph Theory (4-0-0) 4

Contact Hours: 52

Course Learning Objectives (CLOs): The students should learn various aspects and theories on Sets, Logical structures, which are widely used in solving problems by way of computers. Students should develop in depth exposure to aspects of Graph theories & related topics. Also, the students should learn various counting techniques and theories of Relational mathematics.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Solve** problems related to set theory	1	2	
CO-2	**Derive** a logical conclusion by judicious inferences for a given scenario		2	
CO-3	**Build** graphical representation of real life problems and solution.	2		
CO-4	**Apply** the counting principles to real life scenarios	1	2	
CO-5	**Predict** the future probable states through Recurrence relations & Generating functions.	2		12

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3	2.4										1		

Prerequisites:

Fundamental arithmetic

Contents:

1) **Set Theory:** Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams. **6 Hrs.**
2) **Fundamentals of Logic:** Basic connectives, Truth Tables, Tautologies, Logical equivalence. The laws of logic, The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems. **6 Hrs.**
3) **Relations:** Cartesian Products and Relations, Properties of relations, Computer recognition - zero - one matrices and directed graphs. Equivalence Relations and Partitions. **8 Hrs.**
4) **Functions:** Functions - Plain, One-to-one, Onto and Bijective functions in Computer science. **3 Hrs.**
5) **Introduction to Graph Theory:** Directed and Un-directed graphs, Basic Terminologies: Walk, Path, circuit and cycle, Connected and Disconnected graphs, complete graph, Component and Compliment of a graph and Graphs isomorphism. Subgraph: spanning subgraph, Induced subgraph. Vertex Degree: Degree of a vertex, Regular graph, Konigsberg's bridge problem, Eularian graph: Euler trail and circuit. **6 Hrs.**
6) **Planar Graphs:** Planar graphs - Definition and examples. Bipartite and complete Bipartite graphs, Dual of a planar graph, Observations about a planar graph and its dual. Graph Coloring - Properly Coloring of a Graph, Chromatic Number. Kuratowsky's theorem. **7 Hrs.**
7) **Fundamental Principles of Counting:** The Rules of Sum and Product, Permutations, Combinations: The Binomial Theorem, Combinations with Repetition. **6 Hrs.**
8) **The Principles of Inclusion and Exclusion:** The Principles of Inclusion and Exclusion, Generalizations of the Principle, Derangements - Nothing is in its Right Place, Rook Polynomials and Arrangements with Forbidden Position. **5 Hrs.**
9) **Generating Functions:** Introductory examples, Definition and examples: Calculation techniques. **3 Hrs.**
10) **Recurrence Relations:** First order Linear Recurrence Relation – Formulation problems and examples Second-order Linear Homogeneous. Rings and Ring structures. **2 Hrs.**

Reference books:

[1] Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", 5/e, Pearson Education, 2006

[2] Basavaraj S Anami and Venakanna S Madalli: "Discrete Mathematics – A

Concept based approach", Universities Press, 2016
[3] Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", PHI Learning

18UISC303 Unix and Shell Programming (3-0-2) 4

Contact Hours: 52

Course Learning Objectives (CLOs): The objective of the course is to provide a sound technical exposure to the concepts, commands, and procedures in a multiuser, multitasking operating system. To expertise the students with sound knowledge and superior competence in Shell programming in an extremely effective way. They would have a clear appreciation of the role of an O.S. in a computing environment.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Describe** the architecture and features of UNIX Operating System and distinguish it from other Operating System			1
CO-2	**Illustrate** working with vi editor, creating & editing text file with vi editor using standard vi editor commands.		2	
CO-3	**Demonstrate** the execution of various commands by considering the concepts of directories, pipelines,I/O directions, wildcard patterns. basic regular expressions. processes, filters,grep and sed.	2		
CO-4	**Demonstrate** the capabilities to write and execute shell script.	2		1

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	1.0	2.75	2 0											

Contents:

1) **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. **7L+3P=10 Hrs.**

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. **8L+2P=10 Hrs.**
3) **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. 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. **8L+2P=10 Hrs.**
4) **Shell programming.** Ordinary and environment variables. The .profile. Read and read only 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. **8L+2P=10 Hrs.**
5) **The Process**: 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.

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. **8L+4P=12 Hrs.**

Reference books:

[1] Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill
[2] Behrouz A. Forouzan, Richard F. Gilberg: UNIX and Shell Programming-Cengage Learning – India Edition. 2009.
[3] M.G. Venkateshmurthy: “UNIX & Shell Programming”, Pearson Education, 2005.
[4] Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2ndEdition , Wiley,2014.

18UISC304	Computer Organization and Architecture	(3-0-0) 3

Contact Hours: 39

Course Learning Objectives (CLO’s): The course is designed to enable the students to understand the functional units and operation of computer, peripherals and various architectures. The course focuses on the study of basic computer system, concept of programs as a sequence of machine instructions, memory hierarchy, arithmetic and logical operations with integer and floating-point operands pipelining concept and its principles and analysis of system components.

Course Outcomes(Cos):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(1-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Explain** Basic structure and functions of computer components, instruction execution cycle, pipelining, usage of machine instructions, addressing modes, memory operations and various architectures	1		12
CO-2	**Write** assembly language programs for simple tasks		13	14
CO-3	**Apply** various techniques to solve arithmetic involving integer and floating point numbers		1	
CO-4	**Analyze** the performance of various components of computer		2	
CO-5	**Design** memory units for given specifications			3

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	2.5	2.0	1.0									1.0	2.0	1.0

Pre-requisites:

1. Number system and logical design
2. Any computer programming language

Contents:

1) Introduction: Computer types, Functional units: Input unit, Memory unit, Arithmetic & logic unit, Output unit, Control unit, Basic Operational Concepts, Bus Structures, Performance- Processor clock, Basic Performance equation, Pipelining & Superscalar operation, Clock rate, Performance measurement.

05 Hrs.

2) Machine Instructions & Programs: Basic Arithmetic operations ,Memory Locations & Addresses: Byte addressability, Big-endian & Little-endian assignments, Memory Operation, Instruction & Instruction Sequencing: Register Transfer Notation, Basic Instruction Types, instruction execution and straight-line sequencing, condition codes, branching, Addressing Modes, Assembly Language: Assembly Directives, number Notation, stack and subroutine, Logic

Instruction, Shift & Rotate Instructions, Encoding of machine Instructions.

09 Hrs.

3) The Memory System: Some Basic Concepts: Semiconductor RAM Memories: Internal Organization of Memory Chips, Static and Dynamic memories, Structure of larger memories, Read - only Memories: ROM, PROM, EPROM, EEPROM, Flash memory; Speed, Size & Cost: Cache Memories: Mapping functions, Page replacement algorithms, Performance considerations: Interleaving, Hit Rate & Miss Penalty**.** **08 Hrs.**

4) Arithmetic: Arithmetic and Logic Unit, Multiplication of Positive numbers: Signed-Operand Multiplication: Booth Algorithm; Fast Multiplication: Bit-pair Recoding of Multipliers; Integer division: Floating point operations.

07 Hrs.

5) Basic Processing Unit: Some Fundamental Concepts: Register Transfers, Performing an Arithmetic or Logic operation, fetching a Word from Memory, Storing a Word in Memory; Branch instruction: Multiple-Bus Organization.

05 Hrs

6) Pipelining : Basic concepts, Types of hazards, Large Computer Systems-forms of Parallel Processing **05 Hrs.**

Beyond the Syllabus Coverage:

1. Case study : study of architecture of different processors
2. Demonstration of different functional units

Reference Books:

[1] Carl Hamacher, ZvonkoVranesic, Safwat Zaky, “Computer Organization”, 5/e,McGraw-Hill Higher Education , 2011.

[2] William Stallings, “Computer Organization and Architecture”, 9/e, PHI, 2015.

[3] B.Ram, “Computer Fundamentals Architectures and Organizations”, New Age Internationals.

[4] Rajaraman, Computer Organization and Architecture, PHI Learning.

18UISL305	Data Structures Laboratory	(0-0-3) 1.5

Contact Hours: 36

Course Learning Objectives (CLOs): The course is designed to strengthen the analytical and programming skills of students. It introduces the concepts of basic data structure and implementation.

Course Outcomes(COs):

Description of the course outcome:	Mapping to POs(1-12)/PSOs(13-14)

At the end of the course the student will be able to:		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Apply** C programming skills to solve a given problem	2, 13		1, 14
CO-2	**Implement** basic data structures using C programming	2, 13	14	1
CO-3	**Demonstrate** the capabilities to use the data structures in real time applications	2, 13	14	1, 12

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS 13	PSO14
Map ing Level	1.0	3.0										1.0	3.0	1.6

Prerequisites:

1. Background of C programming

Contents: C Programs using following concepts:

1. Single and multi dimensional arrays
2. String manipulation
3. Structure and Union
4. Function and Recursive function
5. Stack, Queue, Circular queue and Double ended queue
6. Linked list: Singly, doubly and circular
7. Tree: Binary tree, Binary Search tree

Reference books:

[1] Y. Langsam, M. J. Augenstein and A. M. Tenenbaum, “Data Structures using C and C++”, 2/e, Pearson Education, 2009.
[2] A. K. Sharma, “Data Structures using C”,2/e, Pearson Education, 2013.
[3] R.S. Saleria, “Data Structures”, Khanna Publishing House
[4] Yashwant Kanetkar. “ Data Structures through C, BPB Publications

18UISL306 Logic Design Laboratory (0 - 0 -3)1.5

Contact Hours: 36

Course Learning Objectives (CLOs): This Laboratory course is designed to strengthen the students to understand the basic concepts of Combinational circuit & Sequential circuit simplification and implementation.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Understand and Verify** truth table of		1,2	

	logic Gates and Boolean functions													
CO-2	**Design** Logic Circuits for given Problem using Knowledge of Boolean, K-map, decoders, multiplexer.						1,13			2,3		12		
CO-3	**Design** & **Implement** the Combinational & Sequential logic circuits						1,13			2,3		12		
PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	2.66	2.0	2.0									1.0	2.0	

Prerequisites:

1. Number system and logical design

List of Experiments:

1. Study and Verify the truth table of various gates
2. Realization of Boolean Functions
   - Simply the given Boolean expression and to realization it using Basic gates and universal gates
   - Realize the adder and subtractor circuits using Basic gates and universal gates
   - Simplify the given Boolean expression using Map Entered variable technique and realize using Multiplexers
   - Implement the given Boolean functions using decoders
   - Design and implement following code conversion using logic gates.
   - Design and implement Two bit comparator using basic gates

3 Flip-Flops:

- To realize and study of Flip –Flop Circuits
- Application of Flip-Flops:
  - Design and implementation of Mod-n synchronous counter
  - Design and implementation of Mod-n asynchronous counter
  - To realize and implement Shift Registers/Ring counter

Reference books:

[1] Thomas Floyd, “Digital Fundamentals”, 3/e, Universal Book Stall, New Delhi. 2011

[2] Leach and Malvino, “Digital Principles and Applications”,6/e, Tata McGraw-Hill,2006

IV Semester

18UMAC400	Engineering Mathematics-IV	(3-0-0) 3

Contact Hours: 39

Course Learning Objectives (CLOs): To provide an insight into applications of conformal mapping, integration of complex functions and application of probability distributions in Engineering.

COs	**Description of the course outcomes:** At the end of course the students will be able to	**Mapping to POs (1-12)**		
		Mastering 3	**Moderate 2**	**Introductory 1**
CO-1	Construct and use the concepts of analytic function to solve the problems arising in Engineering field.			1

CO-2	Utilize conformal transformation and complex integral to transform irregular domain onto a relatively simple domain.		1	
CO-3	Apply discrete and continuous probability distributions in analyzing the probability models arising in engineering field.		1	
CO-4	Make use of the correlation and regression analysis to fit a suitable mathematical model for the statistical data.		1,2	
CO-5	timate the correlation, covariance using joint probability distributions. Also use student's t-distribution, Chi-square distribution as a test of goodness of fit.		1,2	

Course Outcomes (Cos):

POs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mapping Level	1.8	2														

Contents:

1) Calculus of complex functions: Review of function of a complex variable, limits, continuity, and differentiability. Analytic functions: Cauchy-Riemann equations in Cartesian and polar forms. Construction of analytic functions: Milne-Thomson method-Problems. **7 Hrs.**

2) Conformal transformations: Introduction. Discussion of transformations:$w = e^z; w = z^2$, $w = z + \frac{1}{z}, z \neq o)$. Bilinear transformations-Problems. Complex integration: Line integral of a complex function, Cauchy's theorem and Cauchy's Integral theorem. **8 Hrs.**

3) Probability Distributions: Review of basic probability theory. Random variables (discrete and continuous), probability mass/density functions. Binomial, Poisson, exponential and normaldistributions- problems (No derivation for mean and standard deviation)-Illustrative examples. **8 Hrs.**

4) Statistical Methods: Correlation and regression-Karl Pearson's coefficient of correlation and rank correlation-problems. Regression analysis- lines of regression–problems.

Curve Fitting: Curve fitting by the method of least squares- fitting the curves of the form $= ax + b; y = ax^2 + bx + c; y = ax^b$. **8 Hrs.**

5) Joint probability distribution: Joint Probability distribution for two discrete random variables, expectation and covariance.

Sampling Theory: Introduction to sampling distributions, standard error, Type-I and Type-II errors. Test of hypothesis for means, student's t-distribution, Chi-square distribution as a test of goodness of fit. **8 Hrs.**

Text Books:

1) E. Kreyszig:Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed.(Reprint) 2016.

2) B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 44th Ed., 2017.

3) Srimanta Pal et al: Engineering Mathematics, Oxford University Press, 3rd Edition, 2016.

Reference Books:

1. C. Ray Wylie, Louis C. Barrett : "Advanced Engineering Mathematics", 6th Edition, McGraw-Hill Book Co., New York, 1995.
2. S. S. Sastry, Introductory Methods of Numerical Analysis, Prentice Hall of India, 4th Edition 2010.
3. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
4. N. P. Bali and Manish Goyal : A Text Book of Engineering Mathematics, Laxmi Publishers, 7th Ed., 2014.

Web links and Video Lectures:

1. http://nptel.ac.in/courses.php?disciplineID=111.
2. http://www.class-central.com/subject/math(MOOCs).
3. http://academicearth.org/.

18UISC400	Object Oriented Programming	(4-0-0) 4

Contact Hours: 52

Course Learning Objectives (CLOs): Understand fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc. Be aware of the important topics and principles of software development. Have the ability to write a computer program to solve specified problems.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:	Mapping to POs(1-12)/PSOs(13-14)		
	Substantial Level (3)	Moderate Level (2)	Slight Level (1)

CO-1	**Illustrate** the basic OOP concepts, language constructs, generic programming, inheritance, access specifiers and polymorphism in C++	1	12
CO-2	**Apply** C++ language constructs, runtime polymorphism, exception handling, file handling and stream objects to solve a real life problem.	13	2, 3 14
CO-3	**Design** template functions and classes for any given problem	3	13

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3.0	2.0	2.5									1.0	2.5	1.0

Prerequisites:

1. Basic programming skills
2. Procedure-Oriented Programming

Contents:

1) Introduction to C++: A Review of Structures, Procedure- Oriented Programming Systems, Object-Oriented Programming Systems, Comparison of C++ with C, ConsoleInput/Output in C++, Variables in C++, Reference Variables in C++,Function Prototyping, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions. **5 Hrs.**

2) Class and Objects: Introduction to Classes and Objects Member Functions and Member Data, Objects and Functions, Objects and Arrays. Namespaces, Nested Classes. **5 Hrs.**

3) Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation, The set_new _ handler() function. **5 Hrs.**

4) Constructors and Destructors: Constructors, Destructors, The Philosophy of OOPS **5 Hrs.**

5) Inheritance: Introduction to Inheritance, Base Class and Derived Class Pointers, Function Overriding, Base Class Initialization, The Protected Access Specifier, Deriving by Different Access Specifiers, Different Kinds of Inheritance, Order of Invocation of Constructors and Destructors. **6 Hrs.**

6) Virtual Functions and Dynamic Polymorphism: The Need for Virtual Functions, Virtual Functions, The Mechanism of Virtual Functions, Pure Virtual Functions, Virtual Destructors and Virtual Constructors. **6 Hrs.**

7) Stream Handling: Streams, The Class Hierarchy of Handling Streams, Text and Binary Input/Output, Text Versus Binary Files, Text Input/Output, Binary Input/Output, Opening and Closing Files, Files as Objects of the fstream Class, File Pointer, Random Access to Files, Object Input/Output through Member

Functions, Error Handling, Manipulators. **6 Hrs.**

8) Operator Overloading: Operator Overloading, Overloading the Various Operators - Type conversion. **5 Hrs.**

9) Templates: Introduction, Function Templates, Class Templates. **4 Hrs.**

10) Exception Handling: Introduction, C-Style Handling of Error generating Codes, C++ Style Solution - the try/throw/catch Construct. Limitation of Exception Handling **5 Hrs.**

Reference books:

[1] E. Balagurusamy, "OOPs with C++", 7/e, McGraw-Hill, 2014

[2] Abhiram G. Ranade, "An Introduction to Programming through C++", McGrawHill, 2017

[3] Sourav Sahay, "Object-Oriented Programming with C++", 7/e, Oxford University Press,2012

[4] Herbert Schildt, "The Complete Reference C++", 7/e. TMH, 2014

18UISC401 Microcontrollers (4-0-0) 4

Contact Hours: 52

Course Learning Objectives (CLOs): The course is designed to expose the students to the architecture of Microcontroller 8051 and Microprocessor 8085.The course focuses on the study of basic architecture including addressing modes, instructions, memory design. This course also focuses on the interfacing of Microcontroller with external devices using Embedded C Programming skills.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Illustrate the features of embedded systems, architecture of ARM7 and applications	1		
CO-2	Analyze and understand the instruction set and development tools of ARM	1		13
CO-3	Analyze and understand the THUMB state and achieving	2, 13		1
CO-4	competency in assembly		2, 3	

	programming of ARM													
CO-5	Understand the exception, interrupts and interrupt handling schemes													
CO-6	Understand the architectural features of LPC2148 microcontrollers													
PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	2.33	2.5	2.0										2.0	

Prerequisites:

1. Background of computer organization.
2. Exposure to programming in C

Contents:

1. Introduction to Evolution of Microcontroller and Microprocessor **3 Hrs.**
2. ARM Embedded Systems and ARM Processor Fundamentals :The RISC design philosophy, ARM design philosophy, embedded system hardware- AMBA bus protocol, embedded system software- applications. ARM core data flow model, Registers, CPSR-Processor modes, Banked registers. Pipeline- Characteristics. **10 Hrs.**
3. ARM Instruction Set : Fundamentals of ARM instructions, Barrel shifter, Classification and explanation of instructions with examples-Data processing, Branch, Load-store, SWI and Program Status **10 Hrs.**
4. Introduction to THUMB and ARM Programming : Introduction to THUMB, Differences between ARM and THUMB, Register usage in Thumb, ARM Thumb Interworking. General Structure of ARM assembly module, Assembler directives. Simple ALP programs on Arithmetic & logical operations, Factorial, string operation, sorting, searching, and Scan **08 Hrs.**
5. Exception and Interrupt handling schemes: Exception handling- ARM processor exceptions and modes, vector table, exception priorities, link register offsets. Interrupts- assigning interrupts, interrupt latency, IRQ and FIQ exceptions with example- code for enabling and disabling IRQ and FIQ exceptions, Comparison between exception and interrupts. Interrupt handling schemes- nested interrupt handler, non-nested interrupt handler. Basic interrupt stack design. **07 Hrs.**
6. LPC2148 ARM CPU :Salient features, applications, block diagram, memory mapping. Functional features of Interrupt controller, RTC, USB, UART, I2C, SPI, SSP controllers, watch dog timers and other system control units. **07 Hrs.**
7. Peripherals: GPIO, PLL & Timers Features, Register description with example and Applications, 0**6 Hrs.**

8. Case Study of advanced Microcontrollers: Latest microcontrollers used in industries**.** **02 Hrs.**

Reference books:

[1] ARM System Developer's guide –Andrew N. SLOSS, ELSEVIER Publications, ISBN 978-81-8147-646-3, 2016
[2] ARM Assembly Language – William Hohl, CRC Press, ISBN:978-81-89643-04-1
[3] ARM System-on-chip Architecture by Steve Furber, Pearson Education,ISBN978-81- 317-0840-8, 2E,2012
[4] LPC 2148 USER MANUAL
[5] Embedded Systems: A Contemporary Design Tool- James K. Peckol ISBN: 978-0-471- 72180-2 October 2007, 2008
[6] Jonathan W. Valvano – Brookes / Cole, "Embedded Microcomputer Systems, Real Time Interfacing", 1999, Thomas Learning

WEB Links:

http://www.ocfreaks.com/lpc2148-gpio-programming-tutorial/
http://www.ocfreaks.com/lpc214x-pll-tutorial-for-cpu-and-peripheral-clock/
http://www.ocfreaks.com/lpc2148-timer-tutorial/
http://www.ocfreaks.com/lpc2148-pwm-programming-tutorial/
http://www.ocfreaks.com/lpc2148-adc-programming-tutorial/
http://manish4u.com/interfacing-of-dac-arm-lpc2148/
https://iot.electronicsforu.com/expert-opinion/latest-microcontrollers-released-last-12-months/

18UISC402	Finite Automata and Formal Languages	(3-2-0) 4

Contact Hours: 52

Course Learning Objectives (CLOs): This course focuses on Study abstract computing machines, Language representation techniques, regular expressions, grammar constructions and associated theories and tools to realize formal languages, Employ different types of automata machines to solve problems in computing.

Course Outcomes(COs):

Description of the course outcome:	Mapping to POs(1-12)/PSOs(13-14)

At the end of the course the student will be able to:		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	**Construct** the finite automata for the given patterns and Explain the working and design principles.		3	12
CO-2	**Write** the regular expressions for the given patterns and Explain different techniques and principles used.	1		
CO-3	**Verify** the properties of the given Languages using standard procedure and Explain the language properties.		2	
CO-4	**Design** the grammar for the given languages specification and Explain the design principles.	3		
CO-5	**Write** lexical analyzer and parser for the simple programming construct using standard compiler writing tools.	2		
CO-6	**Design and Verify** the pushdown Automata, Turing Machine for the given languages specification and Explain the underlying working principles.	3		

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3.0	2.5	2.66									1.0		

Prerequisites:

1. Discrete mathematical structures that includes set theory, elements of mathematical reasoning, functions and relations.
2. Some high-level programming language (commonly C).
3. Knowledge of data structures and algorithms is an added advantage.

Contents:

1) **Introduction to Finite Automata:** Structural Representation. The central concepts of Automata theory – Alphabet, Strings & Languages. Finite Automata: Introduction, Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Equivalence of NFA and DFA, Applications of Finite automata, FA with Epsilon (€) transitions. **7L+1T=8 Hrs**
2) **Regular Expressions and languages:** Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions. Properties of Regular Languages (RL): Proving Languages not to be Regular. Closure properties of Regular Languages, Decision properties of Regular Languages,

Equivalence and Minimization of Automata. **7L+2T=09 Hrs**

3) **Context-Free Grammars (CFG) and Languages (CFL):** Context- Free Grammars, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages. **7L+1T=8 Hr**

4) **System Applications & Tools:Lex and Yacc:** The Simplest Lex Program, Recognizing Words with Lex, Grammars, Running Lex and Yacc, Lex vs. Hand Written Lexers Using Lex: Regular Expressions, A Word count program, parsing a Command Line , A C Source Code Analyzer Using Yacc: Grammars, A Yacc Parser, the Lexer, Arithmetic Expressions and Ambiguity Parser: The Role of the Parser. **6L+3T=9 Hrs**

5) **Pushdown Automata (PDA):** Definition of Pushdown Automata,The languages of a PDA, Equivalence of PDA's and CFG'S, Deterministic Pushdown Automata. Properties of Context Free Languages: Normal forms for Context Free Grammar, Pumping lemma for Context Free Languages, Closure properties of Context Free languages. **6L+3T=9 Hrs**

6) **Introduction to Turing Machines (TM):** Problems that computer cannot solve, Turing Machine, Programming Techniques for Turing Machine, Extensions to Basic Turing Machine, Restricted Turing Machines, Turing Machines and Computers. **6L+3T=9 Hrs**

Reference books:

[1] Elaine Rich, Automata, Computability and Complexity, 1st Edition,Pearson education,2012/2013

[2] K L P Mishra, N Chandrasekaran , 3rd Edition, Theory of Computer Science, PhI, 2012.

[3] John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education, 2013

[4] Michael Sipser : Introduction to the Theory of Computation, 3rd edition, Cengage learning,2013

[5] John C Martin, Introduction to Languages and The Theory of Computation, 3rd Edition, Tata McGraw –Hill Publishing Company Limited, 2013

[6] Peter Linz, “An Introduction to Formal Languages and Automata”, 3rd Edition, Narosa Publishers, 1998

[7] Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012

[8] C K Nagpal, Formal Languages and Automata Theory, Oxford University press, 2012.

18UISC403	Design and Analysis of Algorithms	(3-0-0) 3

Contact Hours: 39

Course Learning Objectives (CLOs): The objectives of the course is that the student should develop the analytical skills, learn to design and analyze algorithm and determine complexity/efficiency for various algorithms.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Describe computational solution to well known problems like searching, sorting etc			1
CO-2	Estimate the computational complexity of different algorithms		2	
CO-3	Develop an algorithm using appropriate design strategies for problem solving	3		

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
M pping Level	1	2	3											

Prerequisites:

1. Introduction to proofs, and discrete mathematics and probability
2. Data Structures
3. Some programming skills

Contents:

1) **Introduction:** Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental data Structures. **3 Hrs.**
2) **Fundamentals of the Analysis of Algorithm Efficiency:** Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical Analysis of Recursive and Non-recursive algorithms, Examples. **4 Hrs.**
3) **Brute Force:** Selection Sort and Bubble Sort, Sequential Search and String Matching, Exhaustive Search. **4 Hrs.**
4) **Divide-and-Conquer:** Merge sort, Quick sort **4 Hrs.**
5) **Decrease-and-Conquer: Insertion** Sort, Depth First and Breadth First Search,

Topological Sorting **4 Hrs.**

6) **Transform-and-Conquer:** Balanced Search Trees, Heaps and Heap sort. **4 Hrs.**
7) **Space and Time Tradeoffs:** Sorting by Counting, Input Enhancement in String Matching. **4 Hrs.**
8) **Dynamic Programming:**Warshall's and Floyd's Algorithms **4 Hrs.**
9) **Greedy Technique:** Prim's Algorithm, Dijkstra's Algorithm, Huffman Trees. **4 Hrs.**
10) **Coping with the** Branch and- Bound. **Limitations of Algorithm Power:** Backtracking **4 Hrs.**

Reference books:

[1] Anany Levitin, "Introduction to The Design and Analysis of Algorithms", 2/e Pearson Education, 2011
[2] Thomas H. Coremen, Charles E. Leiserson, Ronal L. Rivest and Clifford Stein, "Introduction to Algorithms", 2/e, PHI ,2008.
[3] Horowitz E., Sahani S., Rajasekharan S., "Computer Algorithms",3/e, Galgotia Publications, 2013.
[4] R.C.T. Lee, S.S.Tseng, R.C. Chang and Y.T. Tsai, "Introduction to the Design and Analysis of Algorithms A strategic Approach", 2/e, TMH, 2008

18UISC404 Operating Systems (3-0-0) 3

Contact Hours: 39

Course Learning Objectives (CLOs): Student should identify the concepts, principles and services of operating system, all fundamentals of operating system abstractions and demonstrate them, to explain protection and security requirements of operating systems analyze basic resource management techniques in job and process scheduling compare different memory management techniques and apply concurrency and synchronization techniques to write concurrent programs.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Describe Operating systems and their functions.	1		
CO-2	Analyze Process, IPC, Scheduling, synchronization, storage management	2		
CO-3	Illustrate various operating system algorithms and Apply various techniques to solve real time		2	12

	problems.													
PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3.0	2.5										1.0		

Prerequisites:

1. Computer organization,
2. The different parts of computer system
3. High level languages such as C.

Contents:

1. **Introduction to operating systems & their classification:** What is an operating system, Mainframe systems, Desktop systems, Multiprocessor system, Distributed system, Clustered system, Real time system, Handheld system, Feature migration, Computing environments, Operating system structures: System components, OS Services, System calls, System programs, System structure, Virtual machines. **6 Hrs.**
2. **Process, Inter process Communication, Threads & CPU Scheduling :** Process concept, Process scheduling, Operation on processes, Cooperating processes, Inter process communication. Threads - Overview, Multithreading models, Threading issues, Pthreads, Java threads. CPU scheduling - Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Real time scheduling. **8 Hrs.**
3. **Process Synchronization and handling Deadlocks:** The Critical section problem, Synchronization hardware, Semaphores, Classical problems of synchronization, Critical regions, monitors. Deadlock - System model, Deadlock characterization, Methods for handling deadlocks - Deadlock prevention, deadlock avoidance, Deadlock detection and recovery from deadlock. **10 Hrs.**
4. **Storage Management: Main** memory management - Background, Swapping, Contiguous allocation, Paging, Segmentation, Segmentation with paging. Virtual memory - Background, Demand paging, Process creation, Page replacement algorithms, Allocation of frames, Thrashing. File System interface - File concept, Access methods, Directory structure, Disk scheduling methods, Disk management, Swap space management. **10 Hrs.**
5. **Protection and Security :** Goals of protection , Domain of protection, Access matrix , implementation of access matrix, Revocation of access rights, The security problem, Authentication, Program threats, System threats, Securing systems and facilities, Intrusion detection. **5 Hrs.**

Reference books:

[1] Abraham silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", 9/e, Jhon wiley& Sons, 2012.

[2] Milan Milankovic, "Operating system concepts and design"; 2/e, Mcgrawhill 2008.

18UISL405 Object Oriented Programming Laboratory (0-0-3) 1.5

Contact Hours: 36

Course Learning Objectives (CLOs): The students will understand the basics of Object Oriented Principles, to design C++ classes , methods using a subset of data types and using assignment, method calls, while loops, for loops, and conditionals, understanding and experiencing the importance of Object Oriented 4G languages like C++ in developing complex software systems.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Describe Operating systems and their functions,	1		
CO-2	Analyze Process, IPC, Scheduling, synchronization, storage management	2		
CO-3	Illustrate various operating system algorithms and Apply various techniques to solve real time problems.		2	12

P →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3.0	2.5										1.0		

Prerequisites:

1. Procedure-Oriented Programming
2. Basic programming skills

List of programs:

1) Programs on method overloading, inline functions, namespaces.

2) Programs on Class and Objects Member Functions Objects and Arrays. Nested Classes.

3) Programs on Dynamic Memory Management.

4) Programs on Constructors and Destructors: Constructors, Destructors.

5) Programs on Inheritance.
6) Programs on Virtual Functions and Dynamic Polymorphism.
7) Programs on Stream Handling.
8) Programs on Operator Overloading
9) Programs on Templates.
10) Programs on Exception Handling

Reference books:

[1] Sourav Sahay, "Object-Oriented Programming with C++", 7/e, Oxford University Press,2012
[2] Herbert Schildt, "The Complete Reference C++", 7/e. TMH, 2014
[3] E. Balaguruswamy, "OOPs with C++", 7/e, McGraw-Hill, 2014

18UISL406 Microcontroller Laboratory (0-0-3) 1.5

Contact Hours: 36

Course Learning Objectives (CLOs): The course is designed to strengthen the analytical and programming skills of students through assembly and embedded C programming. The lab provides a platform for the student to develop and debug the problems related to microcontroller applications. Student completing this course will have a framework for developing, implementing and integrating microcontroller based system.

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Illustrate the instruction set of ARM	2, 13	1	
CO-2	Interface external devices and I/O with ARM	2, 13	1, 5	
CO-3	Develop C language programs and library functions for embedded system applications.	2, 13	1, 3, 5	

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	2.0	3.0	2.0		2.0								3.0	

Prerequisites:

1. Basics of assembly language
2. C programming

Contents:

PART-A Conduct the following Study experiments to learn ALP using ARM, Arithmetic and logical operations, Interrupts related operations, Timer related applications.

PART-B Conducting interfacing experiments to learn embedded C for ARM. LCD-interfacing, Stepper Motor Interfacing, Real timer sensors interfacing, 7-segment LED interface

References:

[4] ARM System Developer's guide –Andrew N. SLOSS, ELSEVIER Publications, ISBN 2016

[5] ARM Assembly Language – William Hohl, CRC Press, ISBN:978-81-89643-04-1

[6] http://www.ocfreaks.com/lpc2148-gpio-programming-tutorial/

18UISL407 Introductory Project (0-0-2) 1

Contact Hours: 36

Course Learning Objectives (CLOs):

Introductory project is introduced with an objective of understanding and identifying the community expectation in terms of possible Engineering solutions by applying the fundamental knowledge of basic sciences and basic engineering courses

Course Outcomes(COs):

Description of the course outcome: At the end of the course the student will be able to:		Mapping to POs(1-12)/PSOs(13-14)		
		Substantial Level (3)	Moderate Level (2)	Slight Level (1)
CO-1	Perform literature review for a given topic	1,2		
CO-2	Identify problem from literature review	1,2		
CO-3	Establish objectives and methodology for the problem defined	1,2		
CO-4	Analyze the existing solution for the identified problem	2,3	5,13,14	6,7
CO-5	Prepare a report and present their findings using PPT	10		11

PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO14
Mapping Level	3.0	3.0	3.0		2.0	1.0	1.0			3.0	1.0		2.0	2.0

Prerequisites:

1. Fundamental knowledge of basic sciences and basic engineering courses.

Contents:

The project shall be engineering oriented in terms of problem definition, related literature survey and existing solutions. The team consisting of 10-12 students shall be asked to identify the problems related to community and try to propose a solution. The faculty members handling the courses for that semester shall guide the students. A committee consisting of minimum 3 faculty members shall evaluate at the end for CIE. There is no SEE for introductory project.

SDM College of Engineering & Technology, Dharwad

Odd Semester 2019-20

Academic Calendar for UG Programmes

Sl. No.	Particulars	Date
1	Registration	27-07-2019 to 31-07-2019

2	Induction program for First Semester (Tentative)	01-08-2019 to 14-08-2019
3	Teaching Commences for odd semester except I Sem	01-08-2019
4	Last date for registration with late fee	06-08-2019
5	Teaching Commences for I semester	16-08-2019
6	Display of attendance	16-09-2019
7	Internal Assessment – IA– I	18-09-2019 to 20-09-2019
8	Communication of performance to the parents	26-09-2019
9	Last date to drop the course	27-09-2019
10	Display of attendance	02-11-2019
11	Internal Assessment –IA– II	04-11-2019 to 06-11-2019
12	Students Feedback	11-11-2019 to 15-11-2019
13	Communication of performance to the parents	13-11-2019
14	Last date to withdraw the course	13-11-2019
15	Teacher – Parents Meet	16-11-2019
16	Internal Assessment –IA– III	27-11-2019 to 29-11-2019
17	Last day of teaching for Odd Semester	30-11-2019
18	Final Lab Assessments	03-12-2019 to 10-12-2019
19	Display of consolidated Continuous Internal Evaluation (CIE) & Attendance	05-12-2019
20	Communication of performance to the parents	05-12-2019
21	Semester End Examination	13-12-2019 to 27-12-2019
22	Inter Semester Recess	28-12-2019 to 12-01-2020
23	Declaration of Results	09-01-2020
24	Communication of performance to the parents by putting on website	10-01-2020
25	Makeup SEE for odd semesters	11-01-2020 to 18-01-2020
Commencement of Even Semester :		**13-01-2020**

Dean (Academic Program) **PRINCIPAL**

Academic Calendar (Tentative) for Even Semester 2019-20
B.E. & M.Tech

Sl. No.	Particulars	Date
1	Registration	09-01-2020 to 11-01-2020
2	Commencement of Teaching	13-01-2020
3	Last date for registration with late fee	18-01-2020

4	Display of attendance	18-02-2020
5	Internal Assessment – IA– I	24-02-2020 to 26-02-2020
6	Communication of performance to the parents	03-03-2020
7	Last date to drop the course	04-03-2020
8	Parents Meet	14-03-2020
9	Insignia – 2020	20-03-2020 & 21-03-2020
10	Display of attendance	30-03-2020
11	Internal Assessment – IA– II	01-04-2020 to 03-04-2020
12	Last date to withdraw the course	08-04-2020
13	Communication of performance to the parents	11-04-2020
14	Feedback by Students	20-04-2020 to 25-04-2020
15	Internal Assessment –IA– III	04-05-2020 to 06-05-2020
16	Last day of teaching for Even Semester	06-05-2020
17	Final Lab Assessments	09-05-2020 to 20-05-2020
18	Display of consolidated Continuous Internal Evaluation (CIE) marks & Attendance for 8th semester	09-05-2020
19	Semester End Examination for 8th semester	11-05-2020 to 19-05-2020
20	Display of consolidated CIE marks & Attendance for 2nd, 4th & 6th semesters (Both for UG & PG)	13-05-2020
21	Communication of performance to the parents	14-05-2020
22	Project exam for 8th semester	21-05-2020 to 26-05-2020
23	Semester End Examination for 2nd, 4th & 6th semesters (Both for UG & PG)	22-05-2020 to 05-06-2020
24	Results for 8th semester	30-05-2020
25	Summer vacation	06-06-2020 to 31-07-2020
26	Announcement of Results for 2nd, 4th & 6th semester (Both for UG & PG)	12-06-2020

Supplementary Semester: 12-06-2020 to 27-07-2020

Commencement of next Academic Year 2020 - 21: 01-08-2020

Dean (Academic Program) **PRINCIPAL**

Supplementary Semester Calendar for B.E./M.Tech/MBA – 2020

Sl. No.	Particulars	VII & VIII Sem (B.E.)	I to VI Sem (B.E.), M.Tech & MBA
1	Registration	01-06-2020 to 03-06-2020	06-06-2020 to 08-06-2020
2	Teaching Commences	01-06-2020	12-06-2020
3	Registration with special permission by Principal	04-06-2020	12-06-2020
4	Internal Assessment (IA) – I	13-06-2020 & 15-06-2020	24-06-2020 & 25-06-2020
5	Internal Assessment (IA) – II	25-06-2020 & 26-06-2020	03-07-2020 & 04-07-2020
6	Internal Assessment (IA) – III	10-07-2020 & 11-07-2020	13-07-2020 & 14-07-2020
7	Display of consolidated Continuous Internal Evaluation (CIE) marks & Attendance	13-07-2020	16-07-2020
8	Supplementary SEE	14-07-2020 to 17-07-2020	18-07-2020 to 23-07-2020
9	Declaration of results	22-07-2020	27-07-2020

Dean (Academic Program) **PRINCIPAL**