SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Syllabus for

B.Sc.(Hons) in Computer Science

Batch 2019 -2020 onwards

Vidyagiri, PRASANTHI NILAYAM – 515 134,
Anantapur District, Andhra Pradesh, India
Ph: (08555) 287239,   Fax: 286919

Website: <u>www.sssihl.edu.in</u>   Email: <u>registrar@sssihl.edu.in</u>

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

Syllabus for Three Year B.Sc. (Hons.) in

Three Year Undergraduate B.Sc. Honours in CS Programme

(Effective from 2019 batch)

The B.Sc. Honours in Computer Science is a well-structured 3 year undergraduate programme in accordance with the UGC regulations.

The qualification required for the admission is successful completion of 12th standard **from any stream** with Mathematics as one of the core subjects at 10+2 level and any board satisfying minimum marks prescribed in the Regulations.

The syllabi for the programme have been prepared keeping in view the following:

1) The knowledge level of the students being admitted to the programme: Students with varying background and from various boards are admitted. Uniformity of students from various subjects and boards are taken into account. This has to be done keeping in view the prerequisites of B.Sc. courses.

2) Students after completing this programme may join some post graduate programmes in the field of computer science, like M.Sc. Data Science and Computing.

   Care has been taken that the students are prepared well enough to take the follow up courses at the post B.Sc. level.

In order to achieve the above objectives, the programme consists of courses from many domains of knowledge in the area of core and advanced computing and in general divided into the following categories:

1) The language courses are introduced to train the students in the skill set of writing and speaking coherently and convincingly on a given topic. These skills are absolutely essential these days in the work environment. These courses are given in the first four semesters.

2) Courses such as Awareness and Environment are novel courses that help the student to relate to their cultural roots on one hand and understand the present scenario in the society with all its complexities and composition.

3) Computer Architecture: Courses like “Fundamentals of Computer Organization”, “Digital Circuits and Logic Design” train the students in learning the fundamentals of Computer Architecture.

4) System Software: Courses like “Operating Systems”, and “System Software” impart knowledge in the internals of system software.

5) Theoretical Computer Science: Courses like “Introduction to Programming Languages”, “Data Structures and Algorithm Analysis” lay down the foundations required and the skills needed to develop algorithms and to perform the analysis.

6) Web and Computer Networks: Courses like “Network Essentials”, “Web programming” provide the internals behind the working of computer networks and the World Wide Web.

7) Mathematics: Courses like Foundations in Algebra, Geometry and Calculus, Ordinary differential equations, Probability and Statistics, Linear Algebra, Statistical Inference, Data Mining and Discrete Mathematics lay the mathematical and statistical foundation for learning computer science.

8) Databases: Courses like “Database Management Systems” and “SQL Programming” provide the knowledge of database theory and Practicals.

9) Technology: Electives like “Internet of Things”, “Cyber-Physical Systems”, “Problem Solving with Artificial Intelligence” empower the student to embrace latest technologies.

10) Programming Skills: Students are also given sufficient training in programming languages like C, C++, Java, SQL and Python.

11) There is a software project in the 6th semester. The aim of the project is to expose the students to the real world situation. The student is expected to present the work to an internal committee in the form of project viva-voce. A separate grade of the project is shown in the grade card.

12) The examination system has got 2 components:
    a. Continuous Internal Evaluation(CIE)
    b. End Semester Examination(ESE)

CIE for each subject consists of quizzes, assignments or oral tests. These tests are conducted at regular intervals during the semester. These help the student keep abreast with the pace of the course and to reinforce learning. The variety of tests enables both the students and the teachers to evaluate the learning outcomes and performance in various forms of expression. Each component of CIE has its own role and importance in the overall evaluation.

ESE on each subject is conducted once at the end of the semester which is comprehensive in its nature covering the entire syllabus of the course.

In addition to the above mentioned examinations, there is a course component called seminar in the 3rd and 4th semesters and a comprehensive Viva voce in the 6th semester.

The Seminar is intended to widen the scope of learning beyond the confines of the syllabus. In the Seminar, students are expected to present (written as well as oral) new topics, new inventions, and articles from magazines and research journals related to advanced topics in computing. These presentations are evaluated and their grades are shown separately in the grade card.

The aim of the comprehensive viva-voce is to examine the student's understanding and assimilation of the theoretical content of the courses that the student has undergone in the programme up to that point. The examination is restricted to core courses and practicals. A group of teachers of the department jointly conduct and evaluate this examination. A separate grade for this examination is shown in the grade card.

* * *

Program Objective:

The main objective of the course is to train the candidate on various fundamental areas of both mathematics and computer science and encourage to take up advanced study or research in these fields.

Program Outcome:

The course structure and syllabus provides a strong foundation in the fundamental and advanced areas of Computer science so that the candidate is ready to take up any post graduate programs in the field of computer science, like M.Sc. Computer Science, Data Science.

Syllabus Structure for the Three Year

B.Sc. Honors in Computer Science

(From 2019-20 batch)

DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

The B.Sc.(Hons) in Computer Science Programme Structure consists of Three Parts.

PART-I: LANGUAGES#

(a) General English (four papers offered, one each in the first four semesters)
(b) Another Language (four papers offered, one each in the first four semesters – Any one out of: HINDI / SANSKRIT / TELUGU / ADDITIONAL ENGLISH)

PART-II: CORE SUBJECTS
(Offered in all the six semesters) – Title of the papers are given below in the Scheme of Instruction & Evaluation and the syllabus contents are enclosed.

PART-III: AWARENESS COURSE and ENVIRONMENTAL COURSE ##

a) Awareness Courses – (UAWR) (six papers offered, one each in all the six semesters)
b) Environmental Courses – (UENT) (two papers offered, one each in the first two semesters)

NOTE: The title of the papers and the syllabus contents of Part-I and Part-III are provided separately.

Paper Code	Title of the Paper	Credits	Periods	Modes of Evaluation	Types of Papers	Maximum Marks
Semester I						
UGEN-101	General English-I #	**5**	5	IE1	T	100
	Another Language-I #	**4**	4	IE1	T	100
UCSH-101	Foundations in Algebra and Geometry	**3**	3	IE1	T	100
UCSH-102	Digital Circuits and Logic Design	**3**	3	IE1	T	100
UCSH-103	Introduction to Programming Languages	**3**	3	IE1	T	100
UCSH-	C-Programming Lab-I	**3**	6	I	P	100

104						
UAWR-100	Awareness Course-I: Sai Education for Transformation (Based on Life and Teachings of Bhagawan Baba)	**2**	2	I	T	50
UENT-101	Environment-I ##	**2**	2	I	T	75
		25 credits	**28** hours			**725** Marks

Paper Code	Title of the Paper	Credits	Periods	Modes of Evaluation	Types of Papers	Maximum Marks
Semester II						
UGEN-201	General English-II #	**5**	5	IE1	T	100
	Another Language-II #	**4**	4	IE1	T	100
UCSH-201	Foundations in Calculus	**3**	3	IE1	T	100
UCSH-202	Fundamentals of Computer Organization	**3**	3	IE1	T	100
UCSH-203	Data Structures and Algorithm Analysis in C	**3**	3	IE1	T	100
UCSH-204	C-Programming Lab-II	**3**	6	I	P	100
UAWR-200	Awareness Course-II: Unity of Religions	**2**	2	I	T	50
UENT-201	Environment-II ##	**2**	2	I	T	75
		25 credits	**28** hours			**725** Marks

Paper Code	Title of the Paper	Credits	Periods	Modes of Evaluation	Types of Papers	Maximum Marks
Semester III						
UGEN-301	General English-III #	**5**	5	IE1	T	100
	Another Language-III #	**4**	4	IE1	T	100
UCSH-301	Ordinary Differential Equations	**3**	3	IE1	T	100
UCSH-302	Probability and Statistics	**4**	4	IE1	T	100
UCSH-303	Object Oriented Programming	**3**	3	IE1	T	100
UCSH-304	C++ Programming Lab	**3**	6	I	P	100
UCSH-305	Seminar-I	**1**	1	I	-	50
UAWR-300	Awareness Course-III: Study of Classics – I: Ramakatha Rasa Vahini	**2**	2	I	T	50
		25 credits	28 hours			**700** Marks

Paper Code	Title of the Paper	Credits	Periods	Modes of Evaluation	Types of Papers	Maximum Marks
Semester IV						
UGEN-401	General English-IV #	**5**	5	IE1	T	100
	Another Language-IV #	**4**	4	IE1	T	100
UCSH-401	Linear Algebra	**3**	3	IE1	T	100
UCSH-402	Statistical Inference	**4**	4	IE1	T	100
UCSH-403	Operating System	**3**	3	IE1	T	100
UCSH-404	Java Programming Lab	**3**	6	I	P	100
UCSH-405	Seminar-II	**1**	1	I	-	50
UAWR-	Awareness Course-IV:	**2**	2	I	T	50

400	Study of Classics – II: Bhagawatha Vahini					
		25 credits	**28** hours			**700** Marks

Paper Code	Title of the Paper	Credits	Periods	Modes of Evaluation	Types of Papers	Maximum Marks
Semester V						
UCSH-501	Computer Oriented Numerical Analysis	**4**	4	IE1	T	100
UCSH-502	Data Mining	**4**	4	IE1	T	100
UCSH-503	Network Essentials	**3**	3	IE1	T	100
UCSH-504	Database Management System	**3**	3	IE1	T	100
UCSH-505	Software Engineering	**3**	3	IE1	T	100
UCSH-506	SQL and Python Lab	**3**	6	I	P	100
UCSH-507	Web Programming Lab	**3**	6	I	P	100
UAWR-500	Awareness Course-V: Eternal Values for the changing World	**2**	2	I	T	50
		25 credits	**31** hours			**750** Marks

Paper Code	Title of the Paper	Credits	Periods	Modes of Evaluation	Types of Papers	Maximum Marks
Semester VI						
UCSH-601	Elective - I	3	3	IE1	T	100
UCSH-602	Elective - II	3	3	IE1	T	100
UCSH-603	Introduction to High Performance Computing	3	3	IE1	T	100
UCSH-604	High Performance Computing Lab	3	6	I	P	100
UCSH-605	Advanced Java Lab	3	6	I	P	100
UCSH-606	Software Project	6	9	I	P	200
UCSH-607	Comprehensive Viva-voce	2	-	E1	COV	50
UAWR-600	Awareness Course-VI: Life and its Quest	2	2	I	T	50
		25 credits	**32** hours			**800** Marks
	GRAND TOTAL	**150 credits**	**175 periods**			**4400 marks**

Modes of Evaluation

Indicator	Legend
IE1	CIE and ESE ; ESE single evaluation
IE2	CIE and ESE ; ESE double evaluation
I	Continuous Internal Evaluation **(CIE)** only Note: 'I' does not connote 'Internal Examiner'
E	End Semester Examination **(ESE)** only Note: 'E' does not connote 'External Examiner'
E1	ESE single evaluation
E2	ESE double evaluation

Continuous Internal Evaluation (CIE) & End Semester Examination (ESE)

Types of Papers

Indicator	Legend
T	Theory
P	Practical
V	Viva voce
PW	Project Work
D	Dissertation

PS: Please refer to guidelines for 'Modes of Evaluation for various types of papers', and 'Viva voce nomenclature & scope and constitution of the Viva voce Boards'.

The software project is for 6 Credits and evaluated for 200 Marks. Evaluation of the project is Internal by the project committee appointed by the HoD. The Evaluation method is given below:

1. Evaluation will be done in THREE parts.
2. First Evaluation is within the first FIVE weeks of the semester for 20% of total Marks. The student is to submit a short written report and face an oral examination by the project committee.
3. Second Evaluation component is spread across the semester. The project supervisor will evaluate the progress of the candidate in the project work on a weekly basis. This component is for 30% of total Marks.
4. Third Evaluation is at the end of the semester for 50% of total Marks. The student is to submit a project report and make an oral presentation. Following which there will be an oral examination by the project committee.
5. **Project Committee:** A student doing project must have a *project supervisor* allotted by the HoD. Project supervisor and another member of the Department nominated by the HoD constitute the Project Committee.

List of Electives

ELEC1 - Graph Theory

ELEC2 - Discrete Mathematics

ELEC3 - Internet of Things

ELEC4 - Problem Solving with Artificial Intelligence

ELEC5 - System Software

ELEC6 - Information Retrieval

ELEC7 - Cyber-Physical Systems

ELEC8 - Microprocessor

ELEC9 - Embedded Computing

UCSH-101: Foundations in Algebra and Geometry; Credits:3

Course Objectives:

This course will introduce students to the fundamental concepts of Algebra and Geometry. It is the foundation course, which is essential to understand future courses like Linear Algebra, and Probability & Statistics. The Geometry part of this course is useful to visualize certain geometric objects during this programme in different courses.

Course Outcome:

Develop the skill set to

- convert real life problems into a linear system of equations wherever applicable
- solve the linear system of equations using different strategies
- perform basic operations like addition, multiplication and inverse on matrices
- sketch the graph of the basic functions in mathematics
- perform basic operations on vectors like addition, product (scalar/cross) and triple product
- visualize certain abstract ideas into understandable content using geometry

Unit	**Topic**	**Contents**	**Periods**
1	Linear System	Gaussian Elimination, Gauss–Jordan Method, Row Echelon Form, Reduced Row Echelon Form, Consistency of Linear Systems, Homogeneous Systems, Non-homogeneous Systems	9
2	Matrix Algebra	Addition, Transposition, Multiplication, Properties of Matrix Multiplication, Matrix Inversion, Elementary Matrices and Equivalence, LU Factorization	9
3	Coordinate Geometry	Coordinates, R^n, Line through Two Points, Plane Containing Three Points, Distance and Angle, Polar Coordinates, Area, Hyperplanes	5
4	Solid Geometry	Points and Coordinates, Scalar Product, Cross Product, Scalar Triple Product, Vector Triple Product, Planes, Lines in Space, Isometries of Space, Projections	8
5	Conics	Conic Sections, Conic as Quadratic Curve, Focal Properties	8
		Total	39

References:

1. Matrix Analysis and Applied Linear Algebra by Carl D. Meyer, SIAM Publications. Chapters: 1 (except 1.4 and 1.6), 2, 3 (except 3.8)

2. Geometry by Roger Fenn, Springer undergraduate mathematics series, 1st Edition, 2003. Chapters: 2, 5 (except 5.10), 7 (till 7.3)

UCSH- 102: Digital Circuits and Logic Design; Credits:3

Course Objectives:

To understand the basic principles and techniques for designing combinational and sequential circuits

Course Outcome:

Obtain the skill set to

- Design a combinational circuits viz., adders, subtractors, encoders, decoders, multiplexers demultiplexers, code converters.
- Design Sequential Circuits viz., counters and registers

Unit	**Topic**	**Contents**	**Periods**
1	Boolean Algebra	Basics Laws of Boolean Algebra, Logic Gates, Simplifications of Boolean equations using K-maps, Code Conversion, (Binary, Octal, Hexadecimal), Overview of Gray codes and Excess – 3 codes.	10
2	Arithmetic Circuits	Adder, Subtractor, Parallel binary adder/Subtractor, binary multiplier and divider, Combinational Circuits Multiplexers, De-Multiplexers, decoders, encoders, Design of code converters.	10
3	Flip-Flops	S-R, D, J-K, T, Clocked Flip-flop, Race around condition, Master slave Flip-Flop, Realisation of one flip-flop using other flip-flop. Shift Registers, Serial -in-serial -out, serial -in-parallel -out, parallel -in-serial -out and parallel -in-parallel -out, Bi-directional shift register.	10
4	Counters	Ripple counter, Synchronous Counter, Modulo Counters, Ring Counter, and Twisted Ring Counter. Memory Devices - RAM, ROM, PAL & PLA	10
Total			40

Reference: Malvino Leach, "Digital Principles and Application", TMH, 1999

Chapters: 1 (1.1-1.7), 2(2.1-2.4), 3 (3.1-3.8), 4 (4.1-4.12), 5 (5.7 and 5.8), 8 (8.1-8.8, 8.10), 9 (9.1-9.6), 10 (10.1-10.4)

Suggested Readings:

1. R.L.Tokheim, “Digital Electronics, Principles and Applications”, Tata McGraw Hill, 1999.
2. W.Gothman, “Digital electronics”, PHI.
3. S. Salivahanan & S. Arivyhgan. “Digital circuits and design”, Vikas Publication, 2001
4. Moris Mano, “Digital Logic and Computer Design”, PHI Publications, 2002
5. R. P. Jain, “Modern Digital Electronics”, TMH, 3rd Edition, 2003

UCSH-103: Introduction to Programming Languages; Credits:3

Course Objectives:

This course is designed to introduce students to algorithmic thinking. It introduces the fundamental structures in algorithm design and how to apply these structures in order to develop algorithms to solve specific problems.

Course Outcome:

Develop the skill set to

- think in the algorithmic way to solve certain simple computational problems
- choose the appropriate control structures necessary to solve specific problems
- formulate simple algorithms for arithmetic and logical problems
- test and execute the programs and correct syntax and logical errors
- implement conditional branching, iteration and recursion
- decompose a problem into functions and synthesize a complete program using divide and conquer approach
- use arrays and a few structures to formulate algorithms

Unit	**Topic**	**Contents**	**Periods**
1	Algorithms	Format – Syntax – Methodology and Language of Algorithms – Data – Data Types – Primitive Operations – Variable Expressions	5
2	Decision Structures	If-Then-Else - Nested If's - Different Type Of Loop Structures - Compound Conditions - Case Statement.	6
3	Sub Algorithms	Functions Procedures- Argument- Parameter Correspondence- Recursive Sub Algorithms.	6
4	Composite Data Structures	Vectors (One Dimensional Arrays) - Sorting and Searching of Vectors - Higher Dimensional Arrays – Structures - Arrays of Structures - Introduction to Sequential Files.	6
5	String Manipulation	Character – Information - String Concepts - Operations on Strings.	6
6	Linear Data Structures	Linear List - Pointers - Sequential Storage Structures For Arrays - Stacks – Queues	6
7	Application of Above Topics		5
		Total	40

Reference: Introduction to Computer Science, An Algorithmic Approach, Jean-Paul Tremblay And Richard B. Bunt, McGraw Hill, Second International Edition, (1981).

Chapters: Ch. 1 to Ch.6, Ch. 8 (sec 8.1 to 8.6), Appendix A

UCSH-104: C Programming Lab-I; Credits:3

Course Objectives:

This lab course introduces the basics of the C language to students. It would enable a student to implement algorithms for simple problems as programs in C thereby improving algorithmic thinking.

Course Outcome:

Develop the skill set to

- formulate the algorithms as C programs for simple problems
- choose simple data types and programming constructs to develop a C program
- translate given algorithms to a working and correct program
- ability to correct syntax errors as reported by the compiler
- ability to identify and correct logical errors encountered at run time
- ability to write iterative as well as recursive programs
- ability to represent data in arrays and simple structures

Unit	Topic	Contents	Periods
1	If-else, for & while loops	Basic programs for learning syntax, Maximum of n numbers, Number raised to a power, Factorial, Printing different patterns of ‘*’ based triangles, Reversing a number.	20
2	Recursion	Power, Factorial, GCD, Fibonacci numbers, Sum upto N, Reversing a number using recursion	18
3	Arrays	Linear Search, Binary Search, Selection Sort, Merge Sort, Removing duplicates, Reversing an array, Matrix addition & Multiplication	20
4	Strings	Length of a string, String comparison, Palindrome check, Case changing, Find a pattern, Replace a pattern	20
		Total	78

References:

1. Computer Science - A Structured Programming Approach using C by Behrouz Forouzan & Richard Gilberg, ISBN-13: 978-0534491321, ISBN-10: 9780534491321, 2005.

2. The C Programming Language by Dennis Ritchie, Prentice Hall India Learning Private Limited; 2 edition (1990)

UCSH-201: Foundations in Calculus; Credits:3

Course Objectives:

Calculus provides the tools and methods for analyzing and solving problems of change and motion quantitatively. It introduces mathematical operations of differentiation and integration and develops concepts of central importance in mathematical sciences.

Course Outcome:

Develop the skill set to

- compute the slope of tangent at a point for given curve
- write a differentiable function into a series
- find the maxima and minima of function in both one and two variables
- compute the area under the curve for a given surface
- perform different operators on vectors like gradient, divergence and curl
- compute the volume of a given solid

Unit	Topic	Contents	Periods
0	Prerequisites	Sets, Real valued functions, Graphs of functions, Limits, Continuous and differentiable functions.	4
1	Differential Calculus	Successive differentiation, Leibnitz Theorem, Taylor's theorem with Lagrange's forms of remainders, Expansion of a function of one variable in Taylors and McLaren's infinite series. Maxima and Minima of one variable, Partial Derivatives, Euler's theorem, change of variables, total differentiation, Errors and approximation. Taylor's series in two variables. Maxima and Minima of two or more variables, Jacobians and properties.	13
2	Integral Calculus	Definite integral and its application for area, length and volume. Multiple integrals. Change of order of Integration. Transformation of integral from Cartesian to polar. Applications in areas, volume and surfaces.	13
3	Vector Calculus	Differentiation of vectors, Curves in space, Vector operator del, Gradient, Divergence and Curl. Integration of vectors: Line integral, Surface integral, Green's theorem in the plane, Stoke's theorem. Volume integral. Divergence theorem, Green's theorem.	9
		Total	39

Reference: Higher Engineering Mathematics by Dr. B. S. Grewal, Khanna Publishers, 44th Edition

Chapters: 4 (4.1 – 4.4, 4.14-4.15), 5 (5.1 – 5.11), 6 (6.8 – 6.13), 7 (7.1 – 7.8), 8 (8.1 – 8.17)

UCSH-202: Fundamentals of Computer Organization; Credits:3

Course Objectives:

This course introduces students to the fundamental concepts of computing system. This is to familiarize students to equip with understanding of hardware design including logic design, basic structure and behavior of the essential functional modules of the computer. It also gives them how system interacts to accomplish required processing for the needs of the user.

Course Outcome:

Develop the skill set to:

- To know basics of Instruction Set Architecture (ISA) and organization of Computer components.
- Solve simple boolean algebra problems as related to designing computer logic, through combinational and sequential logic circuits.
- Understand how computers represent and manipulate data and convert between different number systems.

Unit	Topic	Contents	Periods
1	Introduction to Digital Computer Logic	Boolean algebra, Gates and combinatorial logic, Sequential Logic circuits	10
2	Data Representation	Number systems – Physical Representation, Different Bases, Arithmetic in different bases, Numeric conversion between number bases, Hexadecimal numbers. Data Formats – Introduction, Alphanumeric character data. Representing Numerical data – Unsigned binary and binary-coded decimals, Signed integers, Real numbers.	10
3	Overview of Instruction set design and memory operations	Little Man computer – Layout, Operation, Programs, Instruction set, Instruction cycle. CPU & Memory – Components, Registers, Memory unit, Fetch-execute instruction cycle, Buses, Classification of instructions, Instruction word formats, Instruction word requirements and constraints. CPU & Memory design enhancement and implementation – CPU architectures, CPU Features and enhancements, memory enhancements. Instruction addressing modes – Register addressing, Immediate addressing, Indirect addressing, Register Indirect addressing, Indexed addressing, Indirect indexed addressing.	10
4	Input/Output	Characteristics of typical I/O devices, Programmed I/O, Interrupts – Servicing, Uses, Multiple interrupts &	10

		Prioritization, Direct Memory access, I/O Modules	
		Total	**40**
Reference: The Architecture of Computer Hardware and Systems Software: An Information Technology approach, Fourth Edition, Irv Englander, Publisher Wiley, 2009.			
Chapters: Ch 3 (3.0-3.6), Ch 4 (4.0-4.2), Ch 5 (5.0-5.3), Ch 6, Ch 7, Ch 8 (8.0-8.3), Ch 9, 10(10.1,10.2) Supplementary chapter 1 and 3.			

UCSH-203: Data Structures and Algorithm Analysis in C; Credits:3

Course Objectives:

This course is designed to introduce the process of designing data structures to solve fairly complex problems. It imparts the following.

- basic concepts of data structures and algorithms
- searching and sorting techniques
- basic concepts about stacks, queues, lists, trees and graphs
- enable students to write algorithms for solving problems with the help of fundamental data structures

Course Outcome:

Develop the skill set to

- choose the appropriate data structure for solving a specific problem
- choose an efficient algorithm based on space or time complexity
- ability to analyze the algorithms to determine the time and computation complexity
- Implement different search techniques
- Implement different sorting techniques
- Solve multiple problems using Stacks, Queues, linked lists, trees and graphs
- Implement and analyze the solution to determine the time and computation complexity

Unit	Topic	Contents	Periods
1	Algorithm Analysis	Complexity of Algorithm-Space Complexity and Time Complexity, Mathematical background – Model – What to Analyze	8
2	Lists, stacks and Queues	Abstract Data Types (ADT's) – The List ADT – The Queue ADT – The Stack ADT	8
3	Trees	Preliminaries – Binary Trees-ADT of Binary Tree, – The Search Tree ADT – Binary Search Trees – AVL Tree	8
4	Sorting	Preliminaries –Bubble Sort- Insertion Sort – Shell Sort – Merge Sort – Quick Sort.	8
5	Graph Algorithms	Definitions – ADT for Weighted Graph and Unweighted Graph, Topological Sort – Minimal Spanning Tree	8
		Total	40

Reference: Data Structures and Algorithm Analysis in C, Mark Allen Weiss, second edition, Pearson Education (Singapore) Pvt. Ltd., Indian branch, 482 F. I. E. Patparganj, Delhi 110 092, India(1997)

Chapters: Chap 2 (2.1 to 2.3), chap 3, chap 4 (4.1.1, 4.1.2, 4.2.1, 4.2.2, 4.3.1 to 4.3.5, 4.4.1, 4.4.2), chap 7 (7.1, 7.2.1, 7.4 (only the algorithm; 7.4.1 is excluded), 7.6 (only the algorithm; 7.6.1 is excluded), 7.7.1, 7.7.2, 7.7.3, 7.7.4), chap 9 (9.1.1, 9.2, 9.5.1, 9.5.2). Another sorting algorithm namely Selection sort should be briefed to the students.

UCSH-204: C Programming Lab- II; Credits:3

Course Objectives:

This lab course introduces the basics of data structures using C language. It would enable a student to design & implement data structures in C to solve fairly complex problems thereby improving data structure design skills.

Course Outcome:

Develop the skill set to

- Program basic data structures in C.
- use complex data structures like Linked Lists, Trees and Graphs appropriately to solve problems
- implement different searching and sorting techniques

Unit	**Topic**	**Contents**	**Periods**
1	Pointers	Simple programs for learning C Pointer Concepts	8
2	Linked Lists	Creation, Insertion, Deletion, Applications	15
3	Trees	General Tree, binary Tree, Binary Search Tree, AVL Trees, Applications of trees	25
4	Stacks, Queues	Stack using Linked List and arrays, Queues using Linked List and arrays	15
5	Graphs	Adjacency matrix based graph, Adjacency list based graph, Topological sort	15
		Total	78

References:

1. Computer Science - A Structured Programming Approach using C by Behrouz Forouzan & Richard Gilberg.
2. The C Programming Language by Dennis Ritchie
3. Pointers in C by Yashwant Kanetkar

UCSH–301: Ordinary Differential Equations; Credits:3

Course Objectives:

The course in Ordinary Differential Equations is an excellent vehicle for displaying the interrelations between mathematics, engineering, physics, biology, economic and social sciences. The interest in differential equations may be both theoretical and applied mathematics. More precisely, the objectives of this course are:

- understand all of the concepts relating to the order and linearity of ODEs, analytic and computational solution methods for ODEs, and the real-world applications of ODEs
- apply your understanding of the concepts, formulas, and problem solving procedures to thoroughly investigate relevant physical models

Course Outcome:

Obtain the skill set to

- formulate problems from engineering, physics, biology and economics into differential equations and solve them by using different techniques
- solve first order differential equations utilizing the standard techniques for separable, exact, linear, homogeneous, or Bernoulli cases
- find particular solutions when given initial or boundary conditions
- solve higher order linear differential equations using reduction of order, undetermined coefficients, or variation of parameters
- compute the power series solution about an ordinary point and singular point

Unit	**Topic**	**Contents**	**Periods**
1	Differential Equations and Their Solutions	Classification, Types of Solutions, Initial Value Problem and Boundary Value Problem, Existence and Uniqueness of Solution for an Initial Value Problem	7
2	Solutions for First Order Equations	Variable Separable Equation, Linear Equation and Bernoulli Equation, Exact Differential Equation, Integrating Factors, Transformations	10
3	Exact Solutions for Higher Order Linear Equations	Homogeneous Equation with Constant Coefficients, Method Of Undetermined Coefficients, Method of Variation Of Parameters, Cauchy-Euler Equation, Wronskian - Basic Theorems	10
4	Series Solutions For Second Order Linear Equations	Power Series Solutions about an Ordinary Point, Solutions about a Singular Point, Frobenius Method, Bessel’s Equation Of Order Zero	12
		Total	39

Reference: Introduction to Ordinary Differential Equations, Shepley L Ross, John Wiley and Sons Pub (1989), 4th Edn

Chapters: 1, 2, 4, 6

UCSH-302: Probability and Statistics; Credits:4

Course Objectives:

Probability theory is the branch of mathematics that deals with modelling uncertainty. It is important because of its direct application in areas such as genetics, finance and telecommunications. It also forms the fundamental basis for many other areas in the mathematical sciences including statistics, modern optimization methods and risk modelling. This course providing students with a formal treatment of probability and statistics.

Course Outcome:

Develop the skill set to

- solve real world problems by using conditional probability and Bayes' theorem
- identify statistical distributions by observing real data sets
- identify and use the probabilistic model for non-deterministic events

Unit	Topic	Contents	Periods
1	Fundamentals	Sets, operations on sets, venn diagrams, Permutations and Combinations, Describing and Summarizing Data Sets, Chebyshev's Inequality, Normal Data Sets, Correlation	8
2	Elements of Probability	Sample Space, Events, Algebra of Events, Axioms of Probability, Conditional Probability and Bayes' Formula	7
3	Random Variables and Expectation	Random Variables, Types of Random Variables, Jointly Distributed Random Variables, Independent Random Variables, Conditional Distributions, Expectation and its properties, Variance, Covariance, Moment Generating Functions, Chebyshev's Inequality	12
4	Special Random Variables	Bernoulli, Binomial and Poisson Random Variables, Hypergeometric, Uniform, Normal, Exponential Random Variables and Gamma Distributions	10
5	Distributions arising from the Normal	The chi-square distribution, the t-distribution, the F-distribution	6
6	Sampling Distributions	Random sample, Statistics, population mean, population variance, sample Mean, central limit theorem, Sample Variance, distributions of the sample mean, joint distribution of sample mean and sample variance, sampling from a finite population	9
		Total	52

*Real life data sets to be provided to students for implementing lab exercises.

* Two hours to be allocated in the time table for the same.

Reference: Introduction to Probability and Statistics for Engineers and Scientists, Sheldon M Ross, Elsevier Academic Press, 4th Edition.
Chapters: 1, 2, 3, 4 , 5, 6.

Suggested Readings:
John E. Freund's Mathematical Statistics with Applications, Irwin Miller Marylees Miller, Eighth Edition, Pearson Publications

UCSH-303: Object Oriented Programming; Credits:3

Course Objectives:

To understand and implement standard tools and techniques for software development using object oriented approach.

Course Outcome:

Develop the skill set to

- Analyze and design real world problems using OOPS concepts viz: Encapsulation, object identity, inheritance and polymorphism.
- Handle exceptions and write generic programming
- Write simple C++ programs using basic object oriented paradigms

Unit	Topic	Contents	Periods
1	Introduction	Introducing Object-Oriented Approach, complexity of the structure and design of complex systems, Relating to other paradigms (functional, data decomposition). Basic terms and ideas: Abstraction, Encapsulation, Inheritance, Polymorphism, Review of C, Difference between C and C++ - cin, cout, new, delete operators.	10
2	Classes and Objects	Encapsulation, information hiding, abstract data types, Object & classes, attributes, methods, C++ class declaration, State identity and behavior of an object, Constructors and destructors, instantiation of objects, Default parameter value, object types, dynamic memory allocation, Metaclass/abstract classes.	10
3	Inheritance and Polymorphism	Inheritance, Class hierarchy, derivation – public, private & protected, Aggregation, composition versus classification hierarchies, Polymorphism, Categorization of polymorphism techniques, Method polymorphism, Polymorphism by parameter, Operator overloading, Parametric polymorphism	10
4	Generic function	Template function, function name overloading, Overriding inheritance methods, Run time polymorphism, Multiple Inheritance. Files and Exception Handling: Persistent objects, Streams	10

		and files, Namespaces, Exception handling, Generic Classes	
Total			40
Reference: A.R.Venugopal, Rajkumar, T. Ravishanker “Mastering C++”, TMH, 1997.			
Chapters: 1(1.1.-1.18), 2,3,4,5,6,,9,10,11,12,,13,14,15, 16(16.1-16.3), 17(17.1-17.2), 18 (18.1-18.9), 19 (19.1-19.7)			
Suggested Readings: 1. R. Lafore, “Object Oriented Programming using C++”, Galgotia Publications, 2004. 2. James Rumbaugh “Object Oriented Modeling and Design” Prentice Hall Publications, 1991 3. D. Parasons, “Object Oriented Programming with C++”, BPB Publication. 4. Steven C. Lawlor, “The Art of Programming Computer Science with C++”, Vikas Publication. 5. Schildt Herbert, “C++: The Complete Reference”, 4th Ed., Tata McGraw Hill, 1999. 6. Tony Gaddis, Watters, Muganda, “Object -Oriented Programming in C++”, 3rd Ed., Wiley Dreamtech, 2004 7. Grady Booch Object oriented Analysis and Design with applications 8. S. B. Lippman & J. Lajoie, “C++ Primer”, 3rd Edition, Addison Wesley, 2000.			

UCSH-304: C++Programming Lab; Credits:3

Course Objectives:

To analyse a real world problem and code using object oriented concepts of C++ .

Course Outcome:

Develop the skill set to

- Understand and implement object oriented paradigms viz., Encapsulation, object identity, inheritance and polymorphism.
- Develop a mini project.

Unit	Topic	Contents	Periods
1	Classes and Objects	To set and get the values given by the user, calculate the area/perimeter of geometrical shapes, Using friend functions. Static members and functions to count the number of objects created. Using constructors (default, parameterized, overloaded) defined in (local, global and function) and destructors along with their order of invocation in addition / multiplication of complex numbers (passing objects as parameters)by using inline functions. matrix multiplication using dynamic objects.	20
2	Inheritance: Single,Multiple,Hierarchical, Multilevel,Hybrid.	To write sample codes with different access specifiers,visibility modes and various types of inheritance,To use constructors and destructors in all the types of inheritance to understand the order of invocation.	10
3	Polymorphism, Templates,Exception handling, Namespaces,Files	Operator overloading on: arithmetic, logical, assignment, relational, unary, subscripting, new and delete operators, using abstract classes with virtual functions. Developing sample codes for swapping, maximizing two numbers using templates. Handling division by zero exception, creating own namespace. To read and write into a file.	20
4	Mini project	Student academic/ non-academic activities, Banking systems/ Self-reliance departments/ etc., using all the concepts.	28
		Total	78

Reference: A.R. Venugopal, Rajkumar, T. Ravishanker “Mastering C++”, TMH, 1997.

1. Schildt Herbert, “C++: The Complete Reference”, 4th Ed., Tata McGraw Hill, 1999.

UCSH-401: Linear Algebra; Credits:3

Course Objectives:

This course provides an introduction to Linear algebra and its applications to other disciplines.

Course Outcome:

Develop the skill set to

- identify problems that can be converted into a linear system of equations
- apply techniques to convert the identified problems into a linear system
- compute the dimension of a vector space
- compute the invariant subspaces of a vector space for a given linear transformation
- compute the orthonormal basis from given basis
- compute eigenvalues and eigenvectors of a given square matrix
- solve the linear system using efficient techniques

Unit	Topic	Contents	Periods
1	Vector Spaces	Spaces and Subspaces, Four Fundamental Subspaces, Linear Independence, Basis and Dimension, Rank, Linear Transformations, Change of Basis and Similarity, Invariant Subspaces	11
2	Norms, Inner Products, and Orthogonality	Vector Norms, Matrix Norms, Inner-Product Spaces, Orthogonal Vectors, Gram–Schmidt Procedure, Unitary and Orthogonal Matrices, Complementary Subspaces, Range-Null space Decomposition	14
3	Eigenvalues and Eigenvectors	Determinants, Properties of Determinants, Elementary Properties of Eigen systems, Diagonalization by Similarity Transformations, Normal Matrices, Positive Definite Matrices	14
		Total	39

Reference:

Matrix Analysis and Applied Linear Algebra by Carl D. Meyer, SIAM Publications. Chapters: 4 (except 4.6), 5 (5.1 - 5.6, 5.9, 5.10), 6, 7 (7.1 – 7.2, 7.5 – 7.6)

UCSH – 402: Statistical Inference*; Credits:4

Course Objectives:

The main purpose of Statistical Inference is to make an inference about a population based on information contained in a random sample selected from that population, and to provide an associated measure of goodness for the inference. Another purpose is hypothesis testing for mean and variance.

Course Outcomes:

Develop the skill set to

- identify the nature of a population from a sample
- model the relationship between a scalar response and one or more explanatory variables
- represent and statistically analyze data both graphically and numerically
- find an appropriate regression model for a given data

Unit	**Topic**	**Contents**	**Periods**
1	Parameter Estimation	Point Estimates, Maximum Likelihood Estimators, confidence intervals for mean and variance, Estimating the Difference in Means of Two Normal Populations, Confidence Interval of the Mean of the Exponential Distribution	10
2	Hypothesis Testing	Significance Levels, Hypothesis testing: For the Mean of a Normal Population, For the Equality of Means of Two Normal Populations, For the Variance of a Normal Population, For the Mean of a Poisson Distribution	10
3	Regression	Linear regression equation, least squares estimators, distribution of the estimators, statistical inferences about the regression parameters: regression to the mean, prediction interval of a future response, coefficient of determination, the sample correlation coefficient, analysis of residuals, transforming to linearity, weighted least squares, polynomial regression	12
4	Analysis of Variance	One-way , Two-way ANOVA	10
5	Goodness of Fit Tests	Goodness of Fit Tests: When all Parameters are Specified and when Some Parameters are Unspecified, Tests of Independence.	10
		Total	52

* Monte-Carlo simulations to be used for demonstrations in class.
* Real life data sets to be provided to students for implementing lab exercises.
* Two hours to be allocated in the time table for the same.

Reference:

Introduction to Probability and Statistics for Engineers and Scientists, Sheldon M Ross, Elsevier Academic Press, 4th Edition.

Chapters: 7 (up to 7.4), 8, 9 (up to 9.9), 10, 11 (up to 11.5).

Suggested Readings:

John E. Freund's Mathematical Statistics with Applications, Irwin Miller Marylees Miller, Eighth Edition, Pearson Publications

UCSH-403: Operating System; Credits:3

Course Objective:

To give basic idea of an OS and its functionalities. To give an in- depth understanding of CPU and its scheduling, memory management, and device management. This course covers other functionalities of OS such as file systems, virtual memory, disk request scheduling, concurrent processes, deadlocks, security, and integrity.

Course Outcome:

Develop the skill set to

- To become familiar with the working of the general operating system.
- To give perspectives to understand the OS services.
- To know the working of the operating system processes, memory subsystem and file system.

Unit	Topics Periods	Contents	
1	Introduction	What is an Operating System, Simple Batch Systems, Multi-programmed Batches systems, Time-Sharing Systems, Personal -computer systems, Parallel systems, Distributed Systems, Real -Time Systems, Computing Environments, Computer System operation, I/O Structure, Storage structure and Hierarchy, Operating System components and Services- System calls, System Programs, System Structure	5
2	Processes	Process Concept, Process Scheduling, Operation on Processes CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Process Synchronization: Background, The Critical -Section Problem, Synchronization Hardware, Semaphores, Classical Problems of Synchronization	9
3	Deadlocks	System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.	5
4	Memory Management	Background, Logical versus Physical Address space, swapping, Contiguous allocation, Paging, Segmentation, Virtual Memory: Demand Paging, Page Replacement, Page-replacement Algorithms, Performance of Demand Paging, Allocation of Frames, Thrashing, Other Considerations.	10
5	Storage Management	File system Implementation, Directory Implementation, Allocation methods, free space management, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk Reliability	11

		Total	**40**
Reference: Silberschatz and Galvin, "Operating System Concepts", Wiley-India, International Student Version, 8th Ed., 2010			
Chapters: Text 1: 1, 3 (3.1, 3.2, 3.3), 5 (5.1, 5.2,5.3, 5.5), 6 (6.1, 6.2, 6.4, 6.5, 6.6), 7, 8 (8.1, 8.2, 8.3, 8.4, 8.6), 9 (9.2,9.4,9.5,9.6,9.9), 11(11.2-11.5), 12(12.2,12.4-12.6).			

UCSH-404: **Java Programming Lab; Credits:3**

Course Objectives:

This course introduces the basics of Java programming.

Course Outcome:

Develop the skill set to

- Write java code for basic programming.
- Understand and implement object oriented concepts like Abstraction, Encapsulation and Inheritance
- Implement various data structures like List, Tree and Graph
- Understand related concepts in Java like Exceptions, Packages

Unit	**Topic**	**Contents**	**Periods**
1	Primitives and Objects in Java	Simple programs for learning primitive data types and basics of creating, using objects in java	10
2	Loops and Control structures	Simple programs using if-else, for, while, switch-case, break, continue	8
3	Access modifiers and Constructors	Private, default, protected and public access modifiers for fields and methods. Default Constructor, Constructor invocations, static, abstract, final	15
4	Inheritance	Object class, super, Multi-level inheritance, Typecasting, Constructors in Inheritance, Fields and Methods in Inheritance, Overriding, Final & Abstract classes	15
5	Arrays, Exceptions, Packages, Interfaces	1D and 2D arrays in Java, Inheritance in Arrays, Checked and Unchecked Exceptions, try, catch, throw, finally, Different ways to import packages, Class path, Inheritance in Packages, Using interfaces for multiple inheritance	15
6	Data Structures	List, Tree, Graph	15
		Total	78

References:

1. The Java Programming Language by James Gosling
2. Thinking in Java by Bruce Eckel

UCSH – 501: Computer Oriented Numerical Analysis*   Credits:4

Course Objectives:

This course introduces numerical methods to solve several mathematical problems and an analysis of different methods.

Course Outcome:

Develop the skill set to

- identify and apply the numerical methods to solve a mathematical problem
- analyze and choose the best method to solve the mathematical problem efficiently
- familiar with numerical integration and differentiation, numerical solution of ordinary differential equations

Unit	**Topic**	**Contents**	**Periods**
1	Taylor Polynomial	Taylor Polynomial - Remainder Term - Methods of Polynomial Evaluation	4
2	Error and Floating Point Arithmetic	Floating Point Representation of Numbers - Floating Point Arithmetic - Errors - Propagation of Error	8
3	Roots of Algebraic Equations	Bisection Method - Newton Method - Secant Method - Fixed Point Method - Ill-Behaved Root finding Problems - Stability of Roots	8
4	Polynomial Interpolation	Linear and Quadratic Lagrangian Interpolation Formulas - Divided Differences - Newton's Divided Difference Interpolation Formula - Error in Polynomial Interpolation - Spline Interpolation using Linear, Quadratic and Cubic Splines	10
5	Integration And Differentiation	Trapezoidal Rules - Simpson Rules - Error Formulas - Gaussian Integration - Differentiation by Interpolation - Differentiation by the Method of Undetermined Coefficients - Effects of Error in Function Values	11
6	Ordinary Differential Equations	Solvability and Stability of Initial Value Problems of Ordinary Linear First Order Differential Equation - Euler's Method - Convergence Analysis of Euler's Method - Taylor Method - Runge-Kutta Methods	11
		Total	52

* Numerical methods to be implemented in the lab course available in the same semester

Reference:

Elementary Numerical Analysis, Kendall Atkinson & Weimin Han, Wiley India, 3rd Edition: Chapters: 1, 2, 3, 4, 5, 8

Suggested Readings:

Intro. Methods of Numerical Analysis. S.S.Sastry, 5th edition, PHI, LTD, India.

UCSH – 502: Data Mining* Credits: 4

Course Objectives:
This course introduces different strategies to draw useful information from any given data set.

Course Outcome:
Develop the skill set to

- apply techniques to classify data based on different parameters
- analyze data and derive useful information from it

Unit	Topic	Contents	Periods
1	Introduction	Motivating Challenges, The Origins of Data Mining, Data Mining Tasks, Data Attributes and Measurement, Types of Data Sets, Measurement and Data Collection Issues, Data Preprocessing: Aggregation, Sampling, Dimensionality Reduction	4
2	Basic techniques for Classification	Basic techniques for Classification, Decision Trees, Model Overfitting, Evaluating the Performance of a Classifier, Holdout Method, Random Subsampling, Cross-Validation, Bootstrap.	10
3	Advanced Techniques for Classification	Rule-Based Classifier, Nearest-Neighbor classifiers, Bayesian Classifiers, Artificial Neural Network(ANN), Support Vector Machine (SVM), Ensemble Methods: Bias-Variance Decomposition, Bagging, Boosting, The Receiver Operating Characteristic Curve, Class Imbalance problem.	12
4	Association Analysis	Basic Concepts and Algorithms Frequent Item set Generation-The Apriori Principle, Rule Generation in Apriori Algorithm, Alternative Methods for Generating Frequent Item sets: FP-Growth Algorithm	14
5	Cluster Analysis	The Basic K-means Algorithm, Agglomerative Hierarchical Clustering, The DBSCAN Algorithm, Strengths and Weaknesses of DBSCAN.	12
		Total	52

* Real life data sets to be provided to students for implementing lab exercises.
* Two hours to be allocated in the time table for the same.

References: Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Publishers, 2007, [Chap. 1,2,4:4.1-4.5,5:5.1-5.6, 6:6.1-6.6, 8:8.1-8.4].

Suggested Readings:
1. Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, Morgan Kaufmann pub, 2001

2. Data Mining: Practical Machine Learning Tools and Techniques by Ian H. Witten, Eibe Frank, Mark A. Hall, Morgan Kaufmann pub, 2011, 3rd Ed.

UCSH-503: Network Essentials; Credits:3

Course Objective:

- To develop an understanding of modern network architectures from a design and performance perspective. To introduce the student to the major concepts involved in wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs).
- To provide an opportunity to do network programming To provide a WLAN measurement ideas.

Course Outcome : develop the skill set to

- Explain the functions of the different layers of the OSI Protocol.
- Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
- For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component
- For a given problem related TCP/IP protocol developed the network programming.
- Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

Unit	**Topic**	**Contents**	**Periods**
1	Introduction	Network edge, Network core, ISPs and Internet, Protocol Layers and service models, OSI, TCP/IP reference models	4
2	Application Layer	Principles, Web and HTTP, FTP, SMTP, DNS.	8
3	Transport Layer	Services, Multiplexing and demultiplexing, principles of reliable data transfer, connection oriented transport, TCP, Connectionless support ,UDP, Congestion control	8
4	Network Layer and Routing	Addressing, Service models, datagram service virtual circuit service, routing principles, routing algorithms, IP protocol, routing in internet, IPv6, Multicast routing.	10
5	Link Layer and	The Data Link Layer services, Error Detection and	10

	LAN	Correction, Multiple Access protocols, LAN addresses and ARP, Bridges, Routers. Wireless and Mobile Networks: Wireless links, characteristics, CDMA, IEEE 802.11 wireless LANs, Cellular Internet Access, addressing and routing to mobile users, Mobile IP, Handling mobility in cellular networks.	
		Total	40

Reference: A Top Down Approach Featuring the Internet , by Jim Kurose, Keith Ross, 3rd Ed, Pearson Education 2004

Chapters: 1, 2.1 - 2.5, 3, 4.1 - 4.7,5.1 – 5.6, 6.1 – 6.7.

Suggested Readings:

1. Data And Computer Communications, by William Stallings, VII edition, Pearson Education, 2005.
2. Computer Networks by Andrew S. Tanenbaum, IV Ed, Pearson Education 2003
3. Forouzan B A: Data communication and Networking.

UCSH-504: Database Management System; Credits:3

Course Objectives:

To understand different issues involved in the design and implementation of database systems.

Course Outcome:

Develop skill set to

- Draw ER diagrams for a given problem
- Design relational database with normalization
- Handle concurrency control and storage requirements of a database

Unit	**Topic**	**Contents**	**Periods**
1	Introduction	Characteristics of database approach, Purpose of Database Systems-Drawbacks of file System, data models, DBMS architecture and data independence. E-R Modeling: Entity types, entity set, attribute and key, relationships, relation types, roles and structural constraints, weak entities, enhanced E-R and object modeling, Sub Classes:, Super classes, inheritance, specialization and generalization	10
2	File Organization	Indexed sequential access files, implementation using B++ trees, hashing, hashing functions, collision resolution, extendible hashing, dynamic hashing approach	10

		implementation and performance.	
3	Relational Data Model	Relational model concepts, relational constraints, relational algebra. SQL: SQL queries, programming using SQL EER and ER to relational Mapping: Data base design using EER to relational language, The Unified Modelling language UML-features of Good Relational Design.	10
4	Data Normalization	Functional dependencies, Normal form up to 3rd normal form. Concurrency Control: Transaction processing, locking techniques and associated, database recovery, security and authorization. Recovery Techniques, Database Security	10
		Total	40

Reference: R. Elmasri and SB Navathe, “Fundamentals of Database Systems”, Addison Wesley, 4th Ed., 2004

Chapters: 1.1, 1.2, 1.3, 2.1, 2.2, 3.3, 3.4, 3.5, 4.1, 4.2, 5.1, 5.2, 6.2, 7.1, 8.4, 9.1, 10.2, 10.3, 10.4, 13.8, 14.3, 18.1, 18.2, 19.1, 23.1

Suggested Readings:

1. Abraham Silberschatz, Henry Korth, S. Sudarshan, “Database Systems Concepts”, 4th Edition, McGraw Hill, 1997.
2. Jim Melton, Alan Simon, “Understanding the new SQL: A complete Guide”, Morgan Kaufmann Publishers, 1993.
3. A. K. Majumdar, P. Battacharya, “Data Base Management Systems’, TMH, 1996.
4. Bipin Desai, “An Introduction to database Systems”, Galgotia Publications, 1991.

UCSH-505 : Software Engineering;   Credits:3

Course Objectives:

This course introduces students to the general approach of building a software product. It also introduces the various methods of building software products. It also covers the OO approach of software design and other phases.

Course Outcome:

Develop skill set to

- Design a software model for a software product
- Estimate the cost of planning and implementing a software product
- Calculate the software metrics at different phases of development

Unit	**Topic**	**Contents**	**Periods**
1	Introduction	The Problem Domain, S/W Engineering Challenges, S/W Engineering Approach,	2

2	Software Processes	Software Process, Desired Characteristics, Development Process Models, Other Software Processes	3
3	Requirement Analysis	Software Requirements Analysis and Specification: Software Requirements, Problem Analysis, Requirements Specification, Functional Specification with Use Cases, Validation, and Metrics.	5
4	Planning a software project	Planning Process, Effort Estimation, Project Scheduling and Staffing, Software Configuration Management Plan, Quality Plan, Risk Management, Project Monitoring Plan.	5
5	Function oriented design	Design Principles, Module Level Concepts, Design Notation and Specification, Structured Design Methodology, Verification, Metrics.	5
6	Object oriented design	OO Analysis and Design, OO Concepts, Design Concepts, UML, Design Methodology, Metrics.	5
7	Detailed design	Detailed Design, PDL, Verification, Metrics	5
8	Coding	Programming Principles and Guidelines, Coding Process, Refactoring, Verification, Metrics.	5
9	Testing	Testing fundamentals, Black-box Testing, White-box Testing, Testing Process, Defect Analysis and Prevention.	5
		Total	**40**
Reference: An Integrated Approach To Software Engineering by Pankaj Jalote, 3rd Edition, Narosa Publishing House, New Delhi, 2005			
Chapters: 1-1.3, 2.1-2.4, 3.1-3.6, 5.1-5.7, 6.1-6.6, 7.1-7.6, 8.1-8.3, 9.1-9.5,10.1-10.5			
Suggested Readings: Roger. S. Pressman, "Software Engineering - A Practitioner's approach", 7th Edition, MGH higher Education			

UCSH-506 SQL and Python Lab Credits:3

Course Objectives:

It introduces to basic query writing using SQL and MYSQL workbench
Introduce students the basics of Python programming language

Course Outcome:

Develop skill set to

- Write simple DDL statements using MySQL and MySQL Workbench
- Write simple DML statements MySQL and MySQL Workbench
- Write Python code for various applications
- Identify and use various libraries of Python.

Unit	Topic	Contents	Periods
1	Syntax, Variables, Data Type, Operators, If elif else, For, while Loops	Basic programs for learning syntax, counting series of numbers, Interactive programs to calculate factorial, fibonacci series, printing various structures like triangle, squares, rhombus etc.	10
2	Strings, Lists , Tuples, Dictionaries, Sets	Learning different features of strings and data structures lists which resembles arrays of C. Counting the number of vowels in given paragraph, Writing comprehensions for mathematical equations, Operation on these data structures.	15
3	Modules and Classes	How to write modules in python, writing various customized modules for the purpose of reuse. Factorial module, Person module, Calendar module etc. Also introducing the concept of object oriented programming using classes, inheritance, encapsulation, polymorphism.	15
4	SQL DDL (data Definition language)	**Working on MYSQL command prompt as well as MYSQL workbench** : To Create, insert ,delete, drop, alter, set, rename, operations on the databases (at least 6 tables from the databases mentioned) using with keys viz., primary, foreign and basic data types viz., varchar, int, float, date, time etc.	19
5	SQL DML (Data manipulation languages)	**MYSQL command prompt as well as MYSQL workbench** :Using Select, from, where for single as well as multiples tables, Using Aliases, distinct,*,Union, minus and intersect operations, Nested queries, Exists function, use of NULL, Using aggregate functions viz, max, min and avg.	19
		Python + SQL	78

References: Python:

1. Think Python: How to think like a Computer Scientist by Allen Downey, 2nd edition, O'Reilly Publications:(http://www.greenteapress.com/thinkpython/thinkpython.html)
2. Dive into Python by Mark Pilgrim. (http://www.diveintopython.net/)
3. https://www.w3resource.com/python/python-tutorial.php

SQL:

1. MYSQL reference manual 5.7.
2. https://downloads.mysql.com/docs/workbench-en.pdf

UCSH-507; Web Programming Lab;   Credits:3

Course Objectives:

This course introduces various methods of developing web applications

Course Outcome:

Develop the skill set to

- create web pages using HTML code;
- create web applications for client/server communication
- create web server applications using database connectivity.

Unit	Topic	Contents	Periods
1	HTML	HTML pages using all tags; Tables and Forms	12
2	CSS	Box model; various styling options	6

3	Java Scripts	script basics; functions; objects; control statements; event handling; DOM; event handling; AJAX	24
4	Servlets	NETbeans IDs; Servlet creation; request/response methods; session management; database connectivity	18
5	JSP	creating server pages with different tags; database connectivity; MVC model	18
Total			78

References:

1. Web Technologies - Theory and Practice - M Srinivasan Pearson, 2012
2. Web resources

UCSH-603: Introduction to High Performance Computing; Credits: 3
Course Objectives: The main objective is to introduce different frameworks of parallel and distributed computing. It introduces the key concepts of high performance computing in an easy-paced manner. Initially the basic ideas in multi-core computer architectures are explored through simple programming.
Course Outcomes: To develop the skill set to : ● Familiarize with current parallel architectures, ● Design parallel algorithms to exploit inherent parallelism. ● Understand various principles of parallel algorithm design, communication operations in the network of systems.

Unit	Topic	Contents	Periods
1	Introduction	Motivating Parallelism, Scope of Parallel Computing	3
2	Parallel Programming Platforms	Implicit Parallelism: Trends in Microprocessor Architectures, Limitations of Memory System Performance, Dichotomy of Parallel Computing Platforms, Physical Organizations, and Communication cost in Parallel Machines, Routing Mechanisms for Interconnection Networks, Impact of Process-Processor Mapping and Mapping Techniques.	12
3	Principles of Parallel Algorithm Design	Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models.	10
4	Basic Communication Operations	One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction , All-Reduce and Prefix-Sum Operations , Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some	7

		Communication Operations	
5	Applications	Dense Matrix Algorithms - Matrix-Vector Multiplication, Matrix-Matrix Multiplication, Sorting-Issues in Sorting on Parallel Computers, Sorting Networks Bubble Sort and its Variants, Quicksort Algorithm	8
		Total	40

Reference: Introduction to Parallel Computing, Second Edition, Addison Wesley Publisher, 2003 by Ananth Grama, Anshul Gupta George karypis, Vipin Kumar

Chapters: 1, 2, 3, 4, 8 (8.1,8.2), 9 (9.1,9.2, 9.3, 9.4)

Suggested Readings:

1. G. Hager and Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010. ISBN 9781439811924.
2. V. Eijkhout, Edmond Chow, Robert van de Geijn, Introduction to High Performance ScientificComputing, 2016, available online at https://bitbucket.org/VictorEijkhout/hpc-book-and-course

UCSH-604 ; High Performance Computing Lab ; Credits:3

Course Objectives:

This lab course introduces hands on experience on high performance computing environment.

Course Outcome:

To develop skill set to:

- Do’s and Don'ts of coding practices
- Profiling of sequential programs to identify the performance bottleneck spots and identifying the probable spots where parallelism is possible in the given sequential code.
- Learning Basics of OpenMP, shared memory programming
- Learning basics of MPI, distributed memory programming.

Unit	Topic	Contents	Periods
1	Profiling and Analysing the sequential Programs	Identify the performance bottleneck spots of sequential code using profiling tools like gprof and analysing the result. Learning profiling of the given code using various different programs	18
2	OpenMP Programming	Hello world program, OpenMP shared memory programs to parallelize the code and speed up for the enhancement of the performance, example problems: Parallelizing Breadth first search, depth first search, binary search, sorting algorithms like quick sort merge sort, etc.	30
3	MPI programming	Hello world program using mpi commands, broadcasting programs, point to point communication programs like alltoall, one to all etc., matrix multiplications program	30
		Total	78

References:

1. A. Grama, George Karypis, Vipin Kumar and Anshul Gupta. Introduction to Parallel Computing, Pearson 2003
2. https://computing.llnl.gov/tutorials/mpi/exercise.html
3. https://computing.llnl.gov/tutorials/parallel_comp/

UCSH-605; Advanced Java Lab; Credits:3

Course Objectives:
This course introduces to MVC frameworks viz STRUTS2 ,Spring

Course Outcome:
Develop the skills set to

- Develop simple Struts2/Spring application
- Design a MVC application for a given problem

Unit	Topic	Contents	Periods
1	Basics in JSP and Servlets	Installation of Tomcat and working with Eclipse, Running sample servlet and jsp program, introduction to MVC application using servlets and jsp using redirect and request dispatcher methods.	25
2	Struts2 Framework	Installation and framework set up. Simple struts2 application(jar files needed) with clients, strut.xml and action classes, Writing own action classes ,business services and tag, Post method, Value stack and its application	25
3	Developing solution for a mini project	Restaurant problem/ Library book issue problem / hostel purchase problem etc..	28
		Total	78

References: Struts2.x from www.tutorialpoint.com

******

ELECTIVES

ELEC-1: Graph Theory; Credits: 3

Course Objectives:

This course is aimed to cover a variety of different problems in Graph Theory. In this course students will come across a number of theorems and proofs. Theorems will be stated and proved formally using various techniques. Various graphs algorithms will also be taught along with its analysis.

Course Outcome: Develop the skill set to

- Model and solve real-world problems using graphs and trees, both quantitatively and qualitatively.
- apply the basic concepts of mathematical logic
- describe and solve some real time problems using concepts of graph theory

Unit	**Topic**	**Contents**	**Periods**
1	Introduction	Graphs and Graph model, Connected graphs, Multi graphs and Digraphs	4
2	Degree	Degree of a vertex, Regular graph, Degree sequence	5
3	Isomorphism of graphs	Definition of isomorphism, Isomorphism as a relation	4
4	Trees	Bridges, Trees, Minimal Spanning trees	6
5	Connectivity	Cut-vertices, Blocks, Connectivity	6
6	Traversability	Eulerian graph, Hamiltonian graph	5
7	Planarity	Planar graph, Embedding planar graphs on surface	5
8	Coloring	Color Problem, Vertex Coloring	5
		Total	40

Key Text:

Introduction to Graph Theory (reprint) Gary Chartrand, Ping Zhang, Tata McGraw Hill

Chapters: 1 to 6, 9, 10.

ELEC2: Discrete Mathematics; Credits: 3

Course Objectives:

It emphasizes the conceptual understanding, quantitative reasoning, and contemporary applications by maintaining a dynamic balance among theory, applications, modeling, and drill. Students work to develop a deeper understanding of a range of mathematical topics using interactive math tutorials that allow the student to solve math problems interactively. Students will gain some confidence on how to deal with problems which may arrive in computer science in near future.

Course Outcome:

Develop the skill set to

- write and interpret mathematical notation and mathematical definitions
- improve the proof writing skills
- describe and solve some real time problems using concepts of graph theory

Unit	Topic	Contents	Periods
1	Logic	Proposition – Logic Operators – Conditional Statements – Methods of Proof –Mathematical Induction, Introduction to Predicates and Quantifiers,	8
2	Relations and Digraphs	Product set – Partition – Relation – Digraph – Path – Equivalence relation –Transitive closure – Warshal’s algorithms, The Pigeonhole principle,	8
3	Functions and Groups	Functions from computer science – Growth of functions – Permutation functions – Group: Basic concepts of groups, and Semi groups.	8
4	Order Relations and Structure	Partially ordered set – External element – Lattice – Finite Boolean algebra – Functions of Boolean algebra.	8
5	Graphs and Trees	Graphs – Euler paths and circuits – Hamiltonian path and circuits –-Shortest Path problems, Tree – Labeled trees – Tree Searching – Undirected trees – Minimal Spanning trees.	8
		Total	40

Key Text: Kolman, Busby, Ross: Discrete Mathematical Structures, Pearson education, 4th Edition.

Chapters: 2, 3.3, 3.5, 4 to 9 [proofs for Prim’s and Kruskal’s algorithms not included].

Reference Books:

1. Trembly J.P and Manohar R: Discrete Mathematical Structures with applications to Computer Science, TMH Pub. Co. Ltd. 2003.
2. Kenneth H. Roson: Discrete Mathematics and it, Applications”, 5th Edition, TMH Pub. Co. Ltd. 2003.

3. T. Veerarajan: Discrete Mathematics, with Graph Theory and Combinations(2008), Tata McGraw – Hill Publishing Company Ltd, New Delhi
4. Discrete Mathematics, Dr A Singaravelu et al, Meenakshi Agency.

ELEC3: Internet of Things; Credits: 3

Course Objectives:
This course will introduce students to the principles of IoT along with designing the basic IoT tools

Course Outcome:
Develop skill set to

- learn and Implement the basic IoT tools
- use Raspberry Pi for creating various applications

Unit	**Topic**	**Contents**	**Periods**
1	Introduction & Domain Specific IoTs	Physical Design, Logical Design, IoT Enabling Technologies, IoT levels and deployment templates, Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Lifestyle	5
2	M2M and IoT System Management	M2M, Difference between IoT and M2M, SDN & NFV for IoT, Software defined networking, Network Function Virtualization, Need for IoT Systems Management, SNMP, Network Operator Requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG	8
3	IoT Platform Design & IoT Systems Logical Design	IoT Design Methodology, Case study on IoT System for Weather Monitoring, Introduction to Python, Python packages of interest for IoT	7
4	IoT Physical Devices	Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry Pi with Python, Other IoT Devices	8

5	IoT Physical Servers and Cloud Offerings	Cloud storage Models, Communication APIs, WAMP, Xively, Django, Designing a RESTful Web API, Amazon Web Services for IoT, SkyNet IoT Messaging platform	7
6	Case Studies in IoT Design	Home Automation, Cities, Environment, Agriculture, Productivity Applications	5
		Total	40

Reference Text: Internet of Things – A Hands-On Approach by Arshdeep Bahga & Vijay Madisetti,

Chapters: 1 – 9

ELEC4: Problem Solving with Artificial Intelligence; Credits: 3

Course Objectives:
This course will introduce students to different techniques in solving AI problems.

Course Outcome:
Develop skill set to

- Understand different approaches involved in solving AI problem
- Analyze different approaches
- Decide the best approach based on space and time constraints

Unit	**Topic**	**Contents**	**Periods**
1	Problem Solving	Introduction, Traditional Problems, Polya’s five steps, Problem Solving Techniques, Human window criteria & ranking of solutions, Classification	3
2	Missionaries & Cannibals	Background, Choosing an appropriate representation, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Related Problems, Playable Program	3
3	The 12 Coins	Background, Solving a smaller problem, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Playable Program	3

4	Cryptarithms	Background, Problem Solving Techniques, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Related Problems, Playable Program	3
5	The Red Donkey Puzzle	Background, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Related Problems, Playable Program	3
6	The 15 Puzzle	Background, Problem Solving Techniques, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Related Problems, Playable Program	3
7	The Knight’s Tour	Background, Problem Solving Techniques, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Related Problems, Playable Program	3
8	Mastermind	Background, Problem Solving Techniques, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Related Problems, Playable Program	3
9	The Monty Hall Problem	Background, Problem Solving Techniques, Solution, Human Problem Solving, Related Problems	3
10	Rubik’s Cube	Background, Problem Solving Techniques, Solution, Human Problem Solving, Human window analysis of solutions, Best Machine Solution, Playable Program	3
11	The Prisoner’s Dilemma	The Traditional Problem, The iterated Prisoner’s Dilemma, Applications in Diverse areas, Related Problems	3
12	Sudoku	Background, Mathematical Analysis, Problem Solving Techniques and Strategies, A Real life experiment, Algorithms for Computer Solutions, Human window analysis of solutions	4
13	Cryptography	Background, Symmetric Key Encryption, Public Key Encryption, RSA Encryption, Issues regarding RSA Cryptographic Systems	3

Total	40
Reference Text: Artificial Intelligence and Problem Solving by Dr. Danny Kopec, Christopher Pileggi, Davind Ungar & Shweta Shetty, Mercury Learning and Information Chapters: 1 – 13, 15.	

ELEC5: System Software; Credits: 3

Course Objectives:

Objective is to teach students the working of the end to end process of execution of the program. It introduces to system programming, machine architecture, Machine Language, Assembly language. It also introduces to working of the assembler, loader, linker, macro processor in detail. It also gives different perspective to the operating system.

Course Outcomes:

To develop the skill set to:

- Study the architecture of a hypothetical machine, its assembly language, macro language...
- Programming in assembly language.
- Understand the structure and design of assemblers, linkers and loaders.
- Understand the concepts and theory behind the implementation of high level programming languages.

Units	Topic	Contents	Periods
1	Introduction	System Software, Machine Structure	2
2	SIC	Simplified Instruction Set Computer – Overview	3
3	Assemblers	Machine Dependent Assembler Features, Machine Independent Assembler Features, Design Options	7
4	Loaders and Linkers	Basic Loader Functions, Machine dependent Loader Features, Design Options	7
5	Macro Processors	Basic Functions, Machine Independent Macro Processor Features, Design Options	6
6	Compilers	Basic Functions, Machine dependent compiler features	8
7	Operating Systems	Basic Functions, Machine dependent Operating System Features, Machine Independent Operating System Features	7
		Total	40

Key Text: Systems Programming by Leland Beck, III Ed, Pearson Education, 1997
Chapters: 1 (1.1-1.3), 2 (2.1-2.4), 3 (3.1-3.4), 4 (4.1-4.3), 5 (5.1-5.3), 6 (6.1-6.3)

References:

1. Dhamdhere, "Systems Programming and operating systems", TMH, 1996.
2. Donovan, "System Programming". (McGraw-Hill), 1991.

ELEC6: Information Retrieval; Credits: 3

Course Objectives:

- The main objective of this course is to present the basic concepts in information retrieval and more advanced techniques of multimodal based information systems.
- This course will introduce students to the concepts of searching methods in Web. Covers various information retrieval methods.
- Provides an insight into various concepts of IR: Word statistics, Vector space model (relevance feedback, query expansion, document normalization, document re-ranking), evaluation of retrieval, generalized VSM, latent semantic indexing, Web retrieval, data fusion, metasearch, multimodal retrieval, applications

Course Outcome:

Develop skill set to

- Implement index construction and compression.
- Implement basic text classifications.
- understand the underlined problems related to IR and acquired the necessary experience to design, and implement real applications using Information Retrieval systems.

Unit	**Topic**	**Contents**	**Periods**
1	Introduction	Boolean retrieval, The term vocabulary and postings lists, Dictionaries and tolerant retrieval	9
2	Indexing	Index construction, Index compression	6
3	Scoring	Scoring, term weighting & the vector space model, Computing scores in a complete search system	9
4	Evaluation and Query Expansion	Evaluation in information retrieval, Relevance feedback & query expansion	9
5	Classification	Text classification & Naive Bayes, Vector space classification	7
		Total	40

Key Text: Manning, Raghavan and Schutze, Introduction to Information Retrieval, 2009, http://nlp.stanford.edu/IR-book/information-retrieval-book.html

Chapters: 1 to 9, 13, 14

ELEC7: Cyber-Physical Systems; Credits: 3

Course Objectives:

- This course will introduce students to the core concepts of CPS.
- The objective of this course is to develop an exposition of the challenges in implementing a cyber-physical system from a computational perspective, but based equally on the principles of automated control.
- The course aims to expose the student to real world problems in this domain and provide a walk through the design and validation problems for such systems.

Course Outcome:

Develop skill set to

- Solve various problems on Diff. Equations.
- Implement basic hybrid programs.
- differentiate the principles of design and implementation of cyber-physical systems from that of other embedded systems.
- develop a study on various critical domains, including automotive, avionics, railways, healthcare, atomic energy, power, and industrial automation.

Unit	**Topic**	**Contents**	**Periods**
1	Overview of CPS	Introduction, Hybrid vs CPS, Multi-Dynamical Systems, How to learn about CPS, Computational Thinking for CPS, Learning Objectives	3
2	Differential Equations & Domains	Introduction, Differential Equations as models of Continuous physical processes, Meaning of differential equations, Examples, Domains of differential equations, Syntax of Continuous programs, Semantics of Continuous programs	5
3	Choice & Control	Introduction, Hybrid Programs – Discrete Change, Compositions, Decisions, Choices, Tests, Repetitions, Syntax, Semantics, Design	4
4	Safety & Contracts	CPS Contracts, Logical Formulas for Hybrid Programs, Differential Dynamic Logic, CPS Contracts in Logic, Identifying requirements of a CPS	6
5	Dynamical Systems & Dynamic Axioms	Intermediate conditions for CPS, Dynamic Axioms for Dynamical Systems, A Proof of a short bouncing ball	5

6	Truth & Proof	Introduction, Truth & Proof – Sequents, Proofs, Propositional proof rules, Soundness of proof rules, Proofs with Dynamics, Quantifier Proof Rules, Derived Proof Rules, Sequent proof for the single-hop bouncing ball, Real Arithmetic	6
7	Control Loops & Invariants	Control Loops, Induction for loops, Proof of a happily repetitive bouncing ball, Splitting post-conditions into separate cases	6
8	Events & Responses	Need for Control – Events, Event Detection, Event Firing, Event-Triggered Verification, Event-Triggered Control Paradigm, Physics vs Control Distinctions	5
		Total	40
Reference Text: Logical Foundations of Cyber-Physical Systems by Andre Platzer, Springer International Publishing. Chapters: 1 (up to 1.6), 2 (up to 2.8), 3 (up to 3.5), 4 (up to 4.7), 5 (up to 5.5), 6, 7 (up to 7.6), 8			

Note: ELEC8 consists of ELEC8T and ELEC8P offered together against a 3 credit elective option.

ELEC8T: Microprocessor Theory: 2 Credits			
Course Objective: To introduce architecture and instruction set of 8085 microprocessor			
Course Outcome: Develop Skill set to ● Analyze the instruction set 8085 microprocessor. ● Write simple Assembly language programs and analyze them ● Identify addressing modes and pins of 8086 Microprocessor.			
Unit	**Topic**	**Contents**	**Periods**
1	Overview of VLSI technology	Applications of Microprocessors & Embedded systems in daily life	3
2	8085 Microprocessor Architecture & Microcomputer System	Microprocessor Architecture and its Operations, Memory, Input/Output, 8085 MPU. Instruction Classification, Instruction Format. Overview of Instruction cycle, machine cycle, T-states, op-code fetch memory read and memory write; Interrupts;	3
3	Instruction Set of 8085 µP and Assembly Language Programming-I	Data Transfer (8 Bit, 16 Bit, from memory to µp & from µp to memory) Instructions, Arithmetic (8 & 16 Bit) Operations, Arithmetic Operation related to Memory, Logic Operations (Including Rotate & Compare), Branch Operations;	8
4	Assembly Language Programming-II	Counter and Time Delays, Stack, Subroutine, Conditional Call & Return Instructions; BCD to Binary conversions & arithmetic manipulations	6
5	Intel 8086 Microprocessor	Pin Description, Operating Mode, Registers, Interrupts, Addressing modes. Comparison with 8085 microprocessor. Overview of other Microprocessors from Intel, Zilog and Motorola;	6
		Total	26
Key Text: Microprocessor Architecture, Programming and Applications with 8085/8080A – Ramesh S. Gaonkar, Wiley Eastern Limited.			
Chapters: 2,3,4, 6,7,8,9			
Reference Books: 1. Fundamentals of Microprocessors and Microcomputers – B.RAM, Dhanpat Rai Pub. 2. The Intel Microprocessors8086/8080, 186/286, 386, 486, Pentium and Pentium Pro processor Architecture, Programming and Interfacing – Barry R. Brey, PHI			

3. Understanding of 8085/8086 microprocessor and peripheral ICs- S.K.Sen, New Age International Publishers, 2nd Edition 2010.

ELEC8P: Microprocessor Lab: 1 Credit

The following programs will be simulated and tested on microprocessor hardware.

Program No.	**Program Title**	**Periods**
1	Transfer of a block of numbers	2
2	Addition of n 8-bit numbers	1
3	a) Multiplication by repeated addition b) Multiplication by shift and add method	2
4	Sorting to arrange in ascending order	1
5	Delay routine for a specified time	1
6	16-bit arithmetic (Register pair operations)	1
7	BCD to Binary and Binary to BCD Conversion	2
8	BCD Addition, BCD Subtraction, Multiplication	1
9	Programming with few interface kits like Traffic controller, Elevator, music synthesizer, LCD displays etc.	2
	Total	13

Reference Books:

Fundamentals of Microprocessors and Microcomputers – B.RAM, Dhanpat Rai Pub.

ELEC9: Embedded Computing; Credits: 3

Course Objective:

To introduce the concept of Embedded systems.

To make students learn the various programming methods in Embedded systems.

Course Outcome:

Develop Skill set to

- Learn the embedded platform
- Learn embedded programming methods.

Unit	Topic	Contents	Periods
1	An Overview of Embedded Computing	Introduction to embedded systems; Complex systems and microprocessors, embedded system design process. Design example: Model train controller, Instructions Sets: Preliminaries, ARM Processor	10
2	Embedded Hardware Fundamentals	Programming Input and output, Supervisor mode, exceptions and traps, coprocessors, memory system mechanism, cpu performance	10
3	Embedded Software and Platforms	Basic computing platforms, cpu bus, memory devices and systems, designing with computing platforms	8
4	Program, Design and Analysis Exercises	components for embedded systems, models of programs, assembly, linking and loading, compilation techniques, program level performance analysis, optimization, programs. Validation and testing, system design techniques, design methodologies requirement analysis	12
		Total	40

Key Text: Computers as Components: Principles of Embedded Computing System Design by Wayne Wolf, Morgan Kauffman Publishers, 2nd edition, 2008.

Chapters: 1 (1.11.4), 2 (2.12.3), 3(3.13.6), 4 (4.14.5), 5(5.15.7, 5.95.10), 7(7.17.3)

REFERENCE BOOKS:

1. Embedded Realtime Systems Programming by Sriram Iyer and Pankaj Gupta, Tata McGraw
Hill, 2004 (Tenth reprint)

2. Embedded Systems, Raj Kamal, Hill

Pub. Co. Ltd, 11th print 2007. [Chaps 15, Appendix G]

3.An Embedded Software Primer, David E. Simon, Pearson Education, 2007. [Chps 510]

4. Programming for Embedded Systems, Dream Software Team, WILEY dreamtech India Ltd.
2005.

****