SOMAIYA VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering

Syllabus

B. Tech Computer Engineering
(Second Year Semester III and IV)

From
Academic Year 2021-22
(Revision-1)

Approved by FOET 08/05/2021 and AC 28/06/2021
SY B. Tech COMP Revision 1.0

K J Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

It is notified for information of all concerned that the Board of Studies at its meeting held on April 28, 2021; and the subsequent meeting of Faculty of Engineering & Technology held on May 08, 2021 and the Academic Council held on June 28, 2021 amended the syllabus of S.Y. B. Tech Computer Engineering and same be brought in to force from Academic Year 2021-22 with immediate effect.

Preamble

KJSCE as a constituent college of Somaiya Vidyavihar University has the academic flexibility to develop and implement its own curriculum *KJSCE-SVU-2020* with features such as inclusion of choice based Open Elective Courses, Audit Courses, Add on Credit Courses, Exposure Courses, etc. Distinct assessment and evaluation methods are also designed based on focus of individual courses. The outcome of this entire exercises; either by way of student placements or the feedback received from all stakeholders is quite encouraging.

At present, Industry is moving towards Industrial revolution 4.0. Knowing very well that every country's education system forms the basis of its progress and the groundwork for its future, we need to be making engineering graduates equipped to take industrial challenges. A common feature in successful education systems is the balance between tradition and the capacity to be flexible and able to adapt to current social trends. To achieve this, Somaiya Vidyavihar University allows for the undergraduate courses to have a focus on the changing industrial scenario.

Our new revision in syllabus *KJSCE-SVU-2020,* introduced from the academic year 2020-21, has been designed based on the revised guidelines from various accrediting bodies.

The said syllabus is a result of expert advice from members of Board of Studies, Faculty of Engineering & Technology and Academic Council; both having due representation from academia as well as appropriate industries. Subsequently faculty members of the college have put in efforts to document it in the form which has been presented here.

Some of the highlights of the *KJSCE-SVU-2020* syllabus are: Introduction of wide choice for branch specific electives, more number of open or interdisciplinary electives, opportunity for internships, etc. Courses like Object Oriented Programming Methodology, Full Stack Development and Digital Design are designed as laboratory oriented courses and pay more attention to hands-on learning.

Focus of academic processes in KJSCE is such that, by the time student completes the requirements of the degree, he/ she will be able to acquire attributes required for profession as an engineer. Outcomes are defined to acquire these attributes which lead to development of curriculum, pedagogy and assessment tools. These tools need to be updated based on experiences of teacher and learner. Hence teaching -learning -evaluation paradigm is going to be a mix of traditional as well as use of ICT tools. Role of the faculty member changes from tutor to trainer / instructor/ facilitator / mentor based on the outcomes targeted.

For measuring learning outcomes of students, traditional methods like tests, laboratory work and End Semester Examinations (ESE) are implemented. Continuous Assessment (CA) is carried out through tests and internal assessment (IA) like quizzes, case studies, mini projects etc. These IA tools enable the students to develop competencies through solutions discussed, improvisations suggested, feedbacks given by faculty members. Through these assessment methods students get opportunity for reading research papers, presenting ideas and working in a team.

Since the assessments are distributed throughout the term the learning process is continuously monitored and graded.

The Department of Computer Engineering courses focus on thrust areas of Department. These areas are Intelligent System and Data Processing, Network System and Security, Image Analysis and Interpretation and System & Software Engineering.

College promotes co-curricular, extra-curricular activities as well as sports; making life outside classroom exciting and rewarding. What makes these activities very effective is the fact that these do not focus only on winning trophies but try to nurture generic skills such as leadership, effective communication, teamwork etc. which are essential skills for a bright professional career.

Along with my colleagues, I welcome you to Department of Computer Engineering and look forward to lead you towards professional career.

Dr. Deepak Sharma
Head
Department of Computer Engineering

Dr. Shubha Pandit
Principal and Dean
Faculty of Engineering and Technology

Vision

To become a center of excellence in discipline of Computer Engineering for developing technically adept professionals with ethical and leadership qualities in service of society.

Mission

- Provide sound technical foundation in Computer Engineering through comprehensive curriculum and application oriented learning.
- Provide ambience for professional growth and lifelong learning for adapting to challenges in rapidly changing technology
- Inculcate social and ethical values and leadership qualities

Program Educational Outcomes (PEO)

A graduate of Computer Engineering will

PEO1. Solve problems in diverse fields using knowledge of Computer Engineering.

PEO2. Excel in professional career, exhibit leadership qualities with ethics &soft skills.

PEO3. Pursue higher education, research or entrepreneurship, engage in professional development, adapt to emerging technologies.

Program Outcomes (PO)

After successful completion of the program Computer Engineering Graduate will be able to:

PO1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **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.

PO3. **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.

PO4. **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.

PO5. **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.

PO6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, cultural, environmental, health, safety and legal issues relevant to the professional engineering practice; understanding the need of sustainable development

PO7. **Multidisciplinary Competence:** Recognize/ study/ analyze/ provide solutions to real-life problems of multidisciplinary nature from diverse fields

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. 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.

PO11. 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.

PO12. 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.

Program Specific Outcomes (PSO)

PSO1: Design, construct and implement hardware and software based modern Computing / Information systems with varying complexities

PSO2: Demonstrate competence in designing, implementation and maintenance of computer based applications, computer-controlled equipment and networks of intelligent devices.

.

Acronym for category of courses		Acronyms used in syllabus document	
Acronym	**Definition**	**Acronym**	**Definition**
BS	Basic Science Courses	**CA**	Continuous Assessment
ES	Engineering Science	**ESE**	End Semester Exam
HS	Humanities and Social Sciences including Management Courses	**IA**	Internal Assessment
PC	Professional Core Courses	**O**	Oral
PE	Professional Elective courses	**P**	Practical
OE	Open Elective Courses	**P&O**	Practical and Oral
LC	Laboratory Courses	**TH**	Theory
PR	Project	**TUT**	Tutorial
AC	Audit Course	**TW**	Term work
AOCC	Add on Credit Course	**ISE**	In- Semester Examination
AOAC	Add on Audit Course	**CO**	Course Outcome
AVAC	Add on Value Audit Course	**PO**	Program Outcome
EX	Exposure Course	**PSO**	Program specific Outcome
I	Interdisciplinary courses		

Acronyms used for type of Course

Acronym used	Definition
C	Core Course
E	Elective Course
O	Open Elective Technical
H	Open Elective Humanities/Management/SWAYAM-NPTEL
P	Project
L	Laboratory Course
T	Tutorial
X	Exposure course
A	Audit course

Acronyms used in Eight Digit Course code e.g. 116U06C101

Acronym Serially as per code	Definition
1	SVU 2000 First revision
16	College code
U	Alphabet code for type of programme
06	Programme code
C	Type of course
1	Semester I – semester number
01	First course of semester – course serial number It will be XX for the elective/choice based courses

Semester III

Credit Scheme

Course Code	Course Name	Teaching Scheme (Hrs.) Per Week TH – P – TUT	Total (Hrs.) Per week	Credits Assigned TH – P – TUT	Total Credits	Course Category
116U01C301	Integral Transform and Vector Calculus	3 – 0 – 1	4	3 – 0 – 1	4	BS
116U01C302	Data Structures$	3 – 0 – 0	3	3 – 0 – 0	3	PC
116U01C303	Computer Organization & Architecture	3 – 0 – 0	3	3 – 0 – 0	3	PC
116U01C304	Object Oriented Programming Methodology	1 – 0 – 2	3	1 – 0 – 2	3	PC
116U01C305	Discrete Mathematics	3 – 0 – 1	4	3 – 0 – 1	4	PC
116U01L301	Digital Design Laboratory	1 – 2 – 0	3	1 – 1 – 0	2	PC
116U01L302	Data Structures Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
116U01L303	Computer Organization & Architecture Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
116U01L304	Object Oriented Programming Methodology Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
	Total	**14– 8 – 4**	**26**	**14 – 4 – 4**	**22**	
116U01A30X	Audit Course	2 - 0- 0	2	--	--	AC

$- Common with IT Branch

Examination Scheme

Course Code	Course Name	Examination Scheme							
		Marks							
		CA							
		ISE	IA	ESE	TW	O%	P	P&O#	Total
116U01C301	Integral Transform and Vector Calculus	30	20	50	25	-	-		**125**
116U01C302	Data Structures$	30	20	50	-	-	-	-	**100**
116U01C303	Computer Organization & Architecture	30	20	50	-	-	-	-	**100**
116U01C304	Object Oriented Programming Methodology	30	20	50	-	-	-	-	**100**
116U01C305	Discrete Mathematics	30	20	50	25	-	-	-	**125**
116U01L301	Digital Design Laboratory	-	-	-	50	25	-	-	**75**
116U01L302	Data Structures Laboratory	-	-	-	25	-	-	25	**50**
116U01L303	Computer Organization & Architecture Laboratory	-	-	-	25	25	-	-	**50**
116U01L304	Object Oriented Programming Methodology Laboratory	-	-	-	25	-	-	25	**50**
	Total	**150**	**100**	**250**	**175**	**50**	**-**	**50**	**775**
116U01A30X	Audit Course	-	-	-	-	-	-	-	-

$- Common with IT Branch
% Oral examination based on entire theory syllabus of corresponding theory course,
 # based on practical & Syllabus of the corresponding theory course.

Semester IV
Credit and Examination Scheme

Semester IV
Credit Scheme

Course Code	Course Name	Teaching Scheme (Hrs.) Per Week TH – P – TUT	Total (Hrs.) Per week	Credits Assigned TH – P – TUT	Total Credits	Course Category
116U01C401	Probability, Statistics and Optimization Techniques$	3 – 0 – 1	4	3 – 0 – 1	4	BS
116U01C402	Analysis of Algorithms	3 – 0 – 0	3	3 – 0 – 0	3	PC
116U01C403	Relational Database Management Systems	3 – 0 – 0	3	3 – 0 – 0	3	PC
116U01C404	Theory of Automata with Compiler Design	3 – 0 – 1	4	3 – 0 – 1	4	PC
116U01L401	Web Programming Laboratory	0 - 4 - 0	4	0 – 2 – 0	2	PC
116U01L402	Analysis of Algorithms Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
116U01L403	Relational Database Management Systems Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
116U01P401	Mini Project	1 –2 – 0	3	0 – 3 – 0	3	PR
	Total	**13 – 10 – 2**	**25**	**12 – 7 – 2**	**21**	
116U01A401	Audit Course	2 - 0 – 0	2	--	--	AC

$- Common with IT Branch

Examination Scheme

Course Code	Course Name	Examination Scheme							
		Marks							
		CA		ESE	TW	O%	P	P&O#	Total
		ISE	IA						
116U01C401	Probability, Statistics and Optimization Techniques$	30	20	50	25	-	-	-	**125**
116U01C402	Analysis of Algorithms	30	20	50	-	-	-	-	**100**
116U01C403	Relational Database Management Systems	30	20	50	-	-	-	-	**100**
116U01C404	Theory of Automata with Compiler Design	30	20	50	25	-	-	-	**125**
116U01L401	Web Programming Laboratory	-	-	-	50	-	-	50	**100**
116U01L402	Analysis of Algorithms Laboratory	-	-	-	25	-	-	25	**50**
116U01L403	Relational Database Management Systems Laboratory	-	-	-	25	-	-	25	**50**
116U01P401	Mini Project	-	-	-	50	-	-	50^	**100**
	Total	**120**	**80**	**200**	**200**	**-**	**-**	**150**	**750**
116U01A401	Audit Course	-	-	-	-	-	-	-	-

% Oral examination based on entire theory syllabus, # based on practical & the corresponding theory Syllabus $- Common with IT Branch ^Demo based on mini project and viva based on implementation

Course Code	Course Title							
116U01C301	Integral Transform and Vector Calculus							
	TH		P		TUT		Total	
Teaching Scheme(Hrs.)	03		--		01*		04	
Credits Assigned	03		--		01		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	--	125

* Batch wise Tutorial

Course prerequisites:

- Applied Mathematics-I
- Applied Mathematics –II
- Basics of Vector Algebra

Course Objectives

The objective of this course is to introduce different methods of finding Laplace Transform and Inverse Laplace transform of given function. The course also familiarizes students with the concepts of Fourier series, Fourier Integral and Fourier Transform of a given function. The course also disseminates methods to find Z- Transform and Inverse Z- transform of a function. Concepts of Differentiation and Integration of Vector functions with their applications are also explained in this course. Using these methods it will be possible to analyze and interpret a given real life situation and think of possible solutions.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1. Apply Different methods to find Laplace Transform and Inverse Laplace Transform of a function

CO2. Find Fourier series, Fourier Integral and Fourier Transform of functions.

CO3. Apply Different methods to find Z-Transform and Inverse Z- Transform of a function.

CO4. Apply concepts of Gradient, curl and Divergence of a vector function to solve problems.

CO5. Apply concepts of Vector Integration to solve related problems.

Module No.	Unit No.	Details	Hrs.	CO
1	**Laplace Transform**		12	CO 1
	1.1	Definition of Laplace Transform, Laplace Transform of sin(at), cos(at),sinh(at), cosh(at), erf(t), Heavi-side unit step, dirac-delta function, Laplace Transform of periodic function		
	1.2	Properties of Laplace Transform (without proof): Linearity, first shifting theorem, second shifting theorem, multiplication by t , division by t, Laplace Transform of derivatives and integrals, change of scale.		
	1.3	Inverse Laplace Transform: Partial fraction method, convolution theorem, Application of Laplace Transform: Solution of ordinary differential equations		
2	**Fourier Series**		12	CO2
	2.1	Introduction: Definition, Dirichlet's conditions, Euler's formulae, Fourier Series of Functions: Exponential, trigonometric functions, even and odd functions, half range sine and cosine series		
	2.2	Complex form of Fourier series		
	2.3	Fourier Integral, Fourier Transform and Inverse Fourier Transform		
3	**Z-Transform**		4	CO 3
	3.1	Z-transform of standard functions		
	3.2	Properties of Z-transform (without proof): Linearity, change of scale, shifting property, Multiplication by K, Initial and Final value, Convolution theorem		
	3.3	Inverse Z- transform: Binomial expansion and Method of Partial fraction		
4	**Vector Differentiation**		8	CO 4
	4.1	Scalar and vector product of three and four vectors and their properties.		
	4.2	Gradient of scalar point function, divergence and curl of vector point function.		
	4.3	Solenoidal and irrational vector fields.		

5	**Vector Integration**		**9**	**CO 5**
	5.1	Vector Integral: Line integral, Properties of line integral, Surface integral, Volume integrals.		
	5.2	Green's theorem in a plane (without proof) and related problems		
	5.3	Gauss divergence theorem (without proof), Stokes theorem (without proof) and related problems		
		Total	**45**	

Recommended Books:

Sr. No.	**Name/s of Author/s**	**Title of Book**	**Name of Publisher with country**	**Edition and Year of Publication**
1.	B. S. Grewal	*Higher Engineering Mathematics*	Khanna Publications, India	43rd Edition 2014
2.	Erwin Kreyszig	*Advanced Engineering Mathematics*	Wiley Eastern Limited, India	10th Edition 2015
3.	N.P. Bali and Manish Goyal	*A Textbook of Engineering Mathematics*	Laxmi Publications LTD, India	9th Edition 2016
4.	P. N. Wartikar and J. N. Wartikar	*A text book of Applied Mathematics Vol I & II*	Pune Vidyarthi Gruha, India	6th Edition 2012

Term-Work will consist of Tutorials covering the entire syllabus. Students will be graded based on continuous assessment of their term work. At least 2 tutorials will be conducted with the help of Mathematical and Statistical software in the Laboratory.

Course Code	Course Title							
116U01C302	Data Structure[$]							
	TH		P		TUT		Total	
Teaching Scheme(Hrs.)	03		02		--		05	
Credits Assigned	03		01		--		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	25	150

$ Common with IT Branch

Course prerequisites:

- Any Programming Language

Course Objectives

The objective of this course is to introduce different types of data structure and how user can use data structure in software development. The course also familiarizes students with the concepts of advanced data structures such as balanced search trees, hash tables, priority queues, sorting and searching. Students will be master in the implementation of linked data structures such as linked lists and binary trees using any preferable language. Course mainly focuses on choosing the appropriate data structure for a specified application.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1. Explain the different data structures used in problem solving.

CO2. Apply linear data structures in application development.

CO3. Implement Non-Linear Data Structures to solve problems

CO4: Describe concepts of advance data structures like set, map & dictionary.

CO5. Demonstrate sorting and searching methods.

Module No.	Unit No.	Details	Hrs.	CO
1	**Introduction**		**02**	**CO 1**
	1.1	**Introduction to Data Structures** Types of Data Structures, ADT (Abstract data type)		
2	**Linear data structure**		**12**	**CO 2**
	2.1	**Stack**: The Stack as an ADT, Stack operations, Array Representation of Stack, Application of stack – Polish Notation, Recursion and other applications/Case study.		
		#Self Learning – Application of stack evaluation of postfix and prefix expressions.		
	2.2	**Queues**: The Queue as an ADT, Queue operation, Array Representation of Queue, Circular Queue, Priority Queue, Double ended queue, Application of Queues – Simulation and other applications/Case study.		
		#Self Learning - Application of queue in Josephus Problem.		
	2.3	**Linked List:** Introduction, Representation of Linked List, Linked List v/s Array, Implementation of Linked List, Circular Linked List, Doubly Linked List, Linked Representation of Stack & Queue, Application – Polynomial Representation and Addition, other additional applications/Case study.		
		#Self Learning -Sparse matrix addition		
3	**Non-Linear data structures: Trees, Graph**		**16**	**CO 3**
	3.1	**Trees**: Basic trees concept, Binary tree representation, Binary tree operation, Binary tree traversal, Binary search tree implementation, Threaded binary trees. Different Search Trees -AVL tree, Multiway Search Tree, B Tree, B+ Tree and Trie, Applications/Case study of trees.		
		#Self Learning Learning – Trie, splay trees, Red-Black Trees.		
	3.2	**Graph** - Introduction, Graph Terminologies, Representation, Graph Traversals – Depth First Search (DFS) and Breadth First Search (BFS). Applications/Case study of Graphs.		

4	**Non-Linear data structures: Map, Dictionary**		**7**	**CO 3**
	4.1	**Set:** Set ADT, Set Implementation, Partitions with Union-Find operations, Tree based partition Implementation. **Map:** Map ADT, Implementation, Hash Tables Application of Maps **Dictionary** : Dictionary ADT, Implementation,. Application of Dictionaries		
		#Self learning : Dictionary : Dictionary ADT, Implementation, Application of Dictionaries, Exploring case studies on use of map and dictionary		
5	**Searching and Sorting**		**8**	**CO 4 CO 5**
	5.1	**Sorting** : Sort Concept, Sort Stability , Bubble Sort, Shell Sort, Counting Sort		
		#Self learning : Bucket and Radix sort		
	5.2	**Searching** : Search concept, Linear Search, Binary Search, Hashed List Search, Comparison of searching Techniques		
		Total	**45**	

 # Students should prepare all Self Learning topics on their own. Self-learning topics will enable students to gain extended knowledge of the topic. Assessment of these topics may be included in IA and Laboratory Experiments.

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed	*Fundamentals Of Data Structures In C*	University Press	Second Edition
2.	Richard F. Gilberg & Behrouz A. Forouzan	*Data Structures A Pseudocode Approach with C*	CENGAGE Learning	Second edition
3.	Jean Paul Tremblay, Paul G. Sorenson	*An introduction to data structures with applications*	Tata McGraw-Hill Education	Second Edition
4.	Aaron M Tanenbaum Yedidyah Langsam Moshe J Augentstein	*Data structure Using C*	Pearson	Twelfth Impression 2013
5.	Michael T Goodrich Roberto Tamassia David Mount	*Data Structure and Algorithm in C++*	Wiley	First

Course Code	Course Title							
116U01C303	Computer Organization and Architecture							
	TH		P		TUT		Total	
Teaching Scheme(Hrs.)	03		02		--		05	
Credits Assigned	03		01		--		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	25	--	--	150

Course prerequisites: Students should be familiar with basic concepts of computers and their applications.

Course Objectives:

Students will try to:

1. Conceptualize the basics of organization and architecture of a digital computer and the detailed working of the ALU
2. Learn the function of each element of a memory hierarchy and detailed working of the control unit
3. Study various input output techniques and their applications.

Course Outcomes:

After completing this course, students will be able to:

CO1- Describe and define the structure of a computer with buses structure and detail working of the arithmetic logic unit and its sub modules

CO2- Understand the Central processing unit with addressing modes and working of control unit in depth

CO3- Learn and evaluate memory organization and cache structure

CO4- Summarize Input output techniques and multiprocessor configurations

Module No.	Unit No.	Details of Topic	Hrs.	CO
1.0	**Structure of a Computer System**		04	CO1
	1.1	Introduction of computer system and its sub modules, Basic organization of computer and block level description of the functional units. Von Neumann model		
	1.2	Introduction to buses, bus types, and connection I/O devices to CPU and memory, PCI and SCSI		
2.0	**Arithmetic and Logic Unit**		11	CO1
	2.1	Introduction to Arithmetic and Logical unit, Computer Arithmetic: Fixed and Floating point numbers, Signed numbers, Integer Arithmetic, 2's Complement arithmetic		
	2.2	Booth's Recoding and Booth's algorithm for signed multiplication, Restoring division and non-restoring division algorithms		
	2.3	IEEE floating point number representation and operations: Addition. Subtraction, Multiplication and Division. IEEE standards for Floating point representations :Single Precision and Double precision Format		
3.0	**Central Processing Unit**		10	CO2
	3.1	CPU architecture, Register organization, Instruction formats and addressing modes (Intel processor), Basic instruction cycle. Control unit Operation ,Micro operations : Fetch, Indirect, Interrupt , Execute cycle Control of the processor, Functioning of micro programmed control unit, Micro instruction Execution and Sequencing, Applications of Micro programming		
	3.2	RISC v/s CISC processors, RISC and CISC Architecture, RISC pipelining, Case study on SPARC		
4.0	**Memory Organization**.		11	CO3
	4.1	Characteristics of memory system and hierarchy, Main memory, Cache memory principles , Elements of Cache Design		
	4.2	ROM, Types of ROM, RAM, SRAM, DRAM, Flash memory, High speed memories		
	4.3	Cache Memory Organization: Address mapping, Replacement Algorithms, Cache Coherence, MESI protocol, Interleaved and associative memories, Virtual memory, Main memory allocation, Segmentation ,Paging, Secondary storage, RAID levels		

5.0	**I/O Organization**		03	CO4
	5.1	External Devices, I/ O Modules		
	5.2	Programmed I/O, Interrupt driven I/O, DMA		
6.0	**Multiprocessor Configurations**		06	CO4
	6.1	Flynn's classification, Parallel processing systems and concepts		
	6.2	Introduction to pipeline processing and pipeline hazards		
	6.3	Design issues of pipeline architecture, Instruction pipelining: Six Stage instruction pipeline.		
	6.4	8086 Instruction (Arithmetic Instructions, Logical Instructions, Data transfer instructions)		
	Self-Learning; Pin Diagram of 8086, Minimum Mode and Maximum mode with timing diagram			
		Total	**45**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	W.Stallings William	*Computer Organization and Architecture: Designing for Performance*	Pearson Prentice Hall Publication	7th Edition
2.	Hamacher, V. Zvonko, S. Zaky	*Computer Organization*	Tata McGraw Hill Publication	5th Edition
3.	Hwang and Briggs	*Computer Architecture and Parallel Processing*	Tata McGraw Hill Publication	
4.	A. Tanenbaum	*Structured Computer Organization*	Prentice Hall Publication	4th Edition.
5.	John Uffenbeck	*8086/8088 families: Design Programming and Interfacing*	Pearson Education	
6.	Douglas Hall	*Microprocessor and Interfacing*	TMH Publication	

Course Code	Course Title							
116U01C304	Object Oriented Programming Methodology							
	TH		P		TUT		Total	
Teaching Scheme (Hrs./Week)	01		02		02*		05	
Credits Assigned	01		01		02		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	25	150

* Batch wise Tutorial

Course prerequisites:

- Basics of Programming concepts

Course Objectives:

This course will provide the concept of object oriented designing and programming using JAVA and C++. These courses also provide differences in Object oriented programming approach in Java and C++. Students will learn about exception handling, Interfaces, file handling, Inheritance and Multithreading.

Course Outcomes:

At the end of successful completion of the course the student will be able to

CO1. Understand the features of object oriented programming compared with procedural approach with C++ and Java

CO2. Explore arrays, vectors, classes and objects in C++ and Java.

CO3. Implement scenarios using object oriented concepts (Drawing class diagram, relationship between classes).

CO4. Explore the interface, exceptions, multithreading, packages

Module No.	Unit No.	Details	Hrs.	CO
1	**Fundamentals of Object oriented Programming**		**04**	**CO1**
	1.1	Introduction, Procedural Programming Approach, Structured Programming Approach, Modular Programming Approach, OOP Approach		
	1.2	Objects and classes, Data abstraction and Encapsulation, Inheritance and Polymorphism, Runtime polymorphism, Static and Dynamic Binding, Exceptions, Reuse, Coupling and Cohesion, Object Oriented Features of Java and C++. Comparing Object Oriented Concepts with Java and C++		
2	**Class, Object, Method and Constructor**		**08**	**CO1, CO2**
	2.1	Class Object and Method: member, method, Modifier, Selector, iterator, State of an object, instanceof operator, Memory allocation of object using new operator.		
	2.2	Method overloading & overriding, constructor, destructor, Types of constructor (Default, Parameterized, copy constructor with object), Constructor overloading, this, final, super keyword, Garbage collection.		
3	**Arrays String and vectors**		**09**	**CO2**
	3.1	Arrays: Arrays: 1D, 2D, Variable Length array, for-each with Array, Array of objects, Vectors: Vector, ArrayList, Wrapper class. Command line Arguments.		
	3.2	Immutable string ,Methods of String class, String comparison, concatenation, substring, toString method		
	3.3	String-Buffer class, StringBuilder class		
4	**Inheritance and Interface**		**08**	**CO1, CO4**
	4.1	Inheritance Types of Inheritance, Final class, abstract class with constructor, abstract and non-abstract methods, super keyword, Method Overriding.		
	4.2	Interface, final keyword Implementing interfaces, extending interfaces Difference between an Abstract class and an Interface		

5	**Class Diagram**		**06**	**CO1, CO 3**
	5.1	Class Diagram		
	5.2	Implementing Aggregation and Association ,composition ,multiplicity ,Generalization		
6	**Exception Handling & Packages, Multithreading**		**10**	**CO4**
	6.1	Packages: Creating Packages, Using Packages, Access Protection, Predefined packages		
	6.2	Exception handling: Exception as objects, Exception hierarchy, Try catch finally Throw, throws		
	6 .3	Multithreading: Thread life cycle, Multithreading advantages and issues, Simple thread program, Thread synchronization.		
		Total	**45**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	Herbert schildt	*The complete Reference JAVA7*	Tata McGraw-Hill	7th Edition 2017
2.	Kathy Sierra	*Sun Certified Programmer for JAVA*	McGraw-Hill Edition	6th Edition, 2013
3.	Sachin Malhotra,Saurabh Chaudhary	*Programming in JAVA*	Oxford University	2nd Edition, 2013
4.	E Balagurusamy	*Object Oriented Programming in C++*	Tata McGraw Hill	5th Edition, 2011
5.	Grady Booch,James Rumbaugh,Ivar Jacobson	*Unified Modeling Language*	Pearson Education	3rd Edition
6.	Yashwant Kanetkar	*Let us C++*	BPB publications	16th Edition, 2020
7.	Ralph Bravaco,Shai simoson	*Java Programming from the Group up*	Tata McGraw-Hill	McGraw-Hill Edition

Course Code	Course Title							
116U01C305	Discrete Mathematics							
Teaching Scheme	TH		P			TUT		Total
(Hrs./Week)	03		--			01*		04
Credits Assigned	03		--			01		04
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	--	125

* Batch wise Tutorial

Course prerequisites

Basic Mathematics

Course Objectives

The objective of this course is to enable students to think logically and mathematically. It will help them to solve the problems with mathematical reasoning, algorithmic thinking, and modeling.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1: Use various mathematical notations, apply various proof techniques to solve real world problems

CO2: Learn and apply core ideas of Set Theory, Relations & Functions

CO3: Use graphs and their types, to solve the practical examples

CO4: Understand the use of Algebraic Structures and lattice, to solve the problems

Module No.	Unit No.	Details of Topic	Hrs.	CO
1		**Set Theory**	**03**	**CO1**
	1.1	Sets, Venn diagrams, Operations on Sets		
	1.2	Laws of set theory, Power set and Products		
	1.3	Partitions of sets, The Principle of Inclusion and Exclusion		
2		**Logic**	**04**	**CO1**
	2.1	Propositions and logical operations, Truth tables		
	2.2	Equivalence, Implications		
	2.3	Laws of logic, Normal Forms		
	2.4	Predicates and Quantifiers		
	2.5	Mathematical Induction		
3		**Relations, Digraphs**	**09**	**CO2**
	3.1	Relations, Paths and Digraphs		
	3.2	Properties and types of binary relations		
	3.3	Manipulation of relations, Closures, Warshall's algorithm		
	3.4	Equivalence relations		
4		**Posets and Lattice**	**09**	**CO2**
	4.1	Partial ordered relations (Posets) ,Hasse diagram		
	4.2	Lattice, sublattice		
	4.3	Types of Lattice ,Boolean Algebra		
5		**Functions and Pigeon Hole Principle**	**03**	**CO3**
	5.1	Definition and types of functions: Injective, Surjective and Bijective		
	5.2	Composition, Identity and Inverse		
	5.3	Pigeon-hole principle, Extended Pigeon-hole principle		
6		**Graphs and Subgraphs**	**04**	**CO4**
	6.1	Definitions, Paths and circuits, Types of Graphs , Eulerian and Hamiltonian		
	6.2	Planer graphs		
	6.3	Isomorphism of graphs		
	6.4	Subgraph		
7		**Algebraic Structures**	**13**	**CO4**
	7.1	Algebraic structures with one binary operation: semigroup, monoids and groups		
	7.2	Cyclic groups, Normal subgroups,		
	7.3	Hamming Code ,Minimum Distance		
	7.4	Group codes ,encoding-decoding techniques		

	7.5	Parity check Matrix ,Maximum Likelihood		
	7.6	**Mathematics of Cryptography** - Modular Arithmetic, Matrices, Linear Congruence, Ring ,GF Fields		
	# **Self Learning Topic – Function Generators**			
			Total	**45**

 # Students should prepare all Self Learning topics on their own. Self-learning topics will enable students to gain extended knowledge of the topic. Assessment of these topics may be included in IA and Laboratory Experiments.

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1	Kenneth H. Rosen	*Discrete Mathematics and its applications*	Tata McGraw Hill	7th Edition, 2017
2	Bernard Kolman, Robert C. Busby	*Discrete Mathematical Structures*	Pearson	6th Edition, 2017
3	C. L. Liu, D. P. Mohapatra	*Elements of Discrete Mathematics West*	Tata McGraw Hill.	4th Edition, 2012
4	Douglas West	*Graph Theory*	Pearson	2nd Edition, 2017

Course Code	Course Title								
116U01L301	Digital Design Laboratory								
	TH			P		TUT		Total	
Teaching Scheme (Hrs./Week)	1			2		-		3	
Credits Assigned	1			1		-		2	
Examination Scheme	Marks								
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	-	-	-	-	50	25	-	-	75

Course prerequisites:

Basics of Digital Electronics

Course Objectives:

The course introduces the students to the concepts of the design and implementation of digital circuits. Laboratory experiments will be used to reinforce the theoretical concepts discussed in lectures. The student will acquire knowledge of gates, flip flops, registers , counters K-maps. The course also includes use of VHDL in the design, simulation, and testing of digital circuits.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1. Recall basic gates & logic families and binary, octal & hexadecimal calculations and conversions.

CO2. Use different minimization techniques and solve combinational circuits.

CO3. Design synchronous and asynchronous sequential circuits.

CO4. Implement digital networks using VHDL.

***Term work will consist of practical's covering the entire syllabus of Digital Design.**
***Students will be graded based on continuous assessment of their term work**

Module No.	Unit No.	Details	Hrs.	CO
		Digital Design Lab		
1	**Binary Arithmetic and Codes:**		**2**	**CO1**
	1.1	Binary Addition and Subtraction (1's and 2's complement method)		
	1.2	Gray Code, BCD Code, Excess-3 code, ASCII Code		
2	**Basic Digital Circuits & Minimization:**		**4**	**CO2**
	2.1	NOT, AND, OR, NAND, NOR, EX-OR, EX-NOR Gates, NAND-NOR Realization.		
	2.2	Solving problems using theorems and properties of Boolean Algebra		
	2.3	Standard SOP and POS form		
	2.4	Reduction of Boolean functions using Algebraic method, K-map method (2,3,4 Variable)		
3	**Combinational Logic Design:**		**3**	**CO2**
	3.1	Half and Full Adder, Half and Full Subtractor, Four Bit Binary Adder, Four Bit Binary Subtractor (1's and 2's complement method)		
	3.3	Multiplexers and Demultiplexers, Decoders,		
	3.4	One bit, Two-bit ,4-bit Magnitude Comparator		
4	**Sequential Logic Design**		**4**	**CO3**
	4.1	Flip Flops: SR, D, JK and T Flip Flop, Truth Tables and Excitation Tables, Flip-flop conversion		
	4.2	Counters: Design of Asynchronous and Synchronous Counters, UP- DOWN counter.		
	4.3	Shift Registers: SISO, SIPO, PIPO, PISO		
5	**Introduction to VHDL**		**2**	**CO4**
	5.1	Introduction to VHDL, Syntax and Programming to be done only during practical sessions		
		Total	**15**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	R. P. Jain	*Modern Digital Electronics*	Tata McGraw Hill	4th Edition 2009
2.	J. Bhasker	*VHDL Primer*	Pearson Education	3rd Edition 2009
3.	M. Morris Mano	*Digital Logic and computer Design*	PHI	1st Edition 2008
4.	Yarbrough John M	*Digital Logic Applications and Design*	Cengage Learning	1st Edition 2006
5.	Douglas L. Perry	*VHDL Programming by Example*	Tata McGraw Hill	4th Edition 2002

Course Code	Course Title							
116U01L302	Data Structures Laboratory							
	TH		P			TUT		Total
Teaching Scheme(Hrs.)	--		02			--		02
Credits Assigned	--		01			--		01
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	--	--	--	25	--	--	25	50

- **Term-Work will consist of practical covering entire syllabus of "Data Structures" 116U01C302. Students will be graded based on continuous assessment of their term work.**
- **Practical and Oral Examination will consist of practical covering entire syllabus of "Data Structures" 116U01L302.**

Note: The faculty should conduct 8-10 experiments or mini project or case studies based on the above syllabus. The programs should be implemented in any programming Language.

Course Code	Course Title							
116U01L303	Computer Organization & Architecture Laboratory							
	TH		P			TUT		Total
Teaching Scheme(Hrs.)			02			--		02
Credits Assigned			01			--		01
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	--	--	--	25	25	--	--	50

Term work will consist of experiments based on syllabus of 'Computer Organization & Architecture' (116U01C303). Students will be graded based on continuous assessment of their term work. Oral examination will be based on syllabus of 'Computer Organization & Architecture' and laboratory experiments**.**

Note: The faculty should conduct around 10 experiments including case studies/self-study based on the above syllabus. The programs should be implemented in C/C++ programming Language.

Course Code	Course Title						
116U01L304	Object Oriented Programming and Methodology						
	TH		P		TUT		Total
Teaching Scheme(Hrs.)	--		02		--		02
Credits Assigned	--		01		--		01
Examination Scheme	Marks						
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	--	--	--	25	--	--	25	50

Term work will consist of experiments based on the syllabus of 'Object Oriented Programming and Methodology' (116U01C304). Students will be graded based on continuous assessment of their term work. Oral examination will be based on the syllabus of 'Object Oriented Programming and Methodology 'and laboratory experiments**.**

Note: Program should be implemented in C++ and Java programming language**.**

Semester IV
Credit and Examination Scheme

Credit Scheme

Course Code	Course Name	Teaching Scheme (Hrs.) Per Week TH – P – TUT	Total (Hrs.) Per week	Credits Assigned TH – P – TUT	Total Credits	Course Category
116U01C401	Probability, Statistics and Optimization Techniques$	3 – 0 – 1	4	3 – 0 – 1	4	BS
116U01C402	Analysis of Algorithms	3 – 0 – 0	3	3 – 0 – 0	3	PC
116U01C403	Relational Database Management Systems	3 – 0 – 0	3	3 – 0 – 0	3	PC
116U01C404	Theory of Automata with Compiler Design	3 – 0 – 1	4	3 – 0 – 1	4	PC
116U01L401	Web Programming Laboratory	0 - 4 - 0	4	0 – 2 – 0	2	PC
116U01L402	Analysis of Algorithms Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
116U01L403	Relational Database Management Systems Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
116U01P401	Mini Project	1 –2 – 0	3	0 – 3 – 0	3	PR
	Total	**13 – 10 – 2**	**25**	**12 – 7– 2**	**21**	
116U01A401	Audit Course	2 - 0 – 0	2	--	--	AC

$- Common with IT Branch

Examination Scheme

Course Code	Course Name	Examination Scheme							
		Marks							
		CA		ESE	TW	O%	P	P&O#	Total
		ISE	IA						
116U01C401	Probability, Statistics and Optimization Techniques$	30	20	50	25	-	-	-	**125**
116U01C402	Analysis of Algorithms	30	20	50	-	-	-	-	**100**
116U01C403	Relational Database Management Systems	30	20	50	-	-	-	-	**100**
116U01C404	Theory of Automata with Compiler Design	30	20	50	25	-	-	-	**125**
116U01L401	Web Programming Laboratory	-	-	-	50	-	-	50	**100**
116U01L402	Analysis of Algorithms Laboratory	-	-	-	25	-	-	25	**50**
116U01L403	Relational Database Management Systems Laboratory	-	-	-	25	-	-	25	**50**
116U01P401	Mini Project	-	-	-	50	-	-	50^	**100**
	Total	**120**	**80**	**200**	**200**		**-**	**150**	**750**
116U01A401	Audit Course	-	-	-	-	-	-	-	-

% Oral examination based on entire theory syllabus, # based on practical & the corresponding theory Syllabus $- Common with IT Branch ^Demo based on mini project and viva based on implementation

Course Code	Course Title							
116U01C401	Probability, Statistics and Optimization Techniques							
	TH		P		TUT		Total	
Teaching Scheme(Hrs.)	03		--		01*		04	
Credits Assigned	03		--		01		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	--	125

* Batch wise Tutorial

Course prerequisites:

- Basics of Statistics and Probability
- Introductory Linear programming problems

Course Objectives

This course Exposes students to the concepts of Correlation, Regression for given bivariate data. Students are made familiar with different discrete and continuous probability distributions. The course acquaints students with concepts of Large sample test, Small sample test and Chi – Square test. The course familiarizes students with different methods of solving Linear and Nonlinear Programming problems. Some basic queuing theory models are also discussed in the course. Using these methods it will be possible to analyze and interpret a given real life situation and think of possible solutions.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1. Apply concepts of correlation, regression for given bivariate data.

CO2. Apply concepts of Binomial, Poisson, Exponential and Normal distribution to solve Engineering problems.

CO3. Apply Large sample test and small sample test to analyze collected data.

CO4. Apply concepts of Linear and Nonlinear programming methods to solve problems.

CO5. Apply the methods of single server limited queue and single server unlimited queue models to analyze and interpret the data.

Module No.	Unit No.	Details	Hrs.	CO
1	**Probability and Probability Distribution**		12	CO 1
	1.1	Conditional Probability, Bayes' theorem, Joint Probability		
	1.2	Discrete and Continuous Probability Distribution		
	1.3	Binomial Distribution, Poisson Distribution		
	1.4	Uniform Distribution, Normal Distribution, Exponential Distribution		
2	**Correlation and Regression**		06	CO 2
	2.1	Correlation, Co-variance, Karl Pearson Coefficient of Correlation & Spearman's Rank Correlation Coefficient.		
	2.2	Regression Coefficients, lines of regression & logistic regression		
3	**Sampling Theory**		07	CO 3
	3.1	Sampling distribution. Test of Hypothesis. Level of significance, critical region. One tailed and two tailed tests. Interval Estimation of population parameters. Large and small samples.		
	3.2	Difference between sample mean and population means for large samples, Test for significance of the difference between the means of two large samples.		
	3.3	Student's t-distribution: Test for significance of the difference between sample mean and population means, Test for significance of the difference between the means of two Samples, paired t-test.		
	3.4	Chi-square distribution as a Test of Independence, Test of the Goodness of fit and Yate's correction.		
	3.5	Fisher's z-test		
4	**Optimization Techniques**		13	CO 4
	4.1	Types of solution, Standard and Canonical form of LPP, Basic and feasible solutions, simplex method.		
	4.2	Artificial variables, Big –M method (method of penalty).		
	4.3	Duality and Dual Simplex method		

	4.4	Unconstrained optimization, problems of two or three variables with one equality constraint using Lagrange's Multiplier method.		
	4.5	Problems of two or three variables with one inequality constraint using Kuhn-Tucker conditions		
5	**Queuing Theory**		**07**	**CO5**
	5.1	Introduction, Features of Queuing, solution of Queuing models. M/M/1(Singal Server, Unlimited Queue Model)		
	5.2	M/M/1 Singal Server, limited Queue Model		
		Total	**45**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	B. S. Grewal	*Higher Engineering Mathematics*	Khanna Publications, India	43rd Edition 2014
2.	Erwin Kreyszig	*Advanced Engineering Mathematics*	Wiley Eastern Limited, India	10th Edition 2015
3.	J. K. Sharma	*Operation research: Theory and Applications*	Laxmi Publications, India	6th Edition 2017
4.	S.C.Gupta and V.K.Kapoor	*Fundamentals of Mathematical Statistics*	Sultan Chand & Sons	11th Edition 2009
5.	Ronald E. Walipole, Raymond H.Myers	*Probabilities & Statistics for Engineers & Scientists*	Pearson Education	9th Edition 2010
6.	P. N. Wartikar and J. N. Wartikar	*A textbook of Applied Mathematics Vol I & II*	Pune Vidyarthigruha, India	6th Edition 2012

Term-Work will consist of Tutorials covering the entire syllabus. Students will be graded based on continuous assessment of their term work. At least 2 tutorials will be conducted with the help of Mathematical and Statistical software in the Laboratory.

Course Code	Course Title							
116U01C402	Analysis of Algorithms							
	TH		P		TUT		Total	
Teaching Scheme (Hrs./Week)	03		02		--		05	
Credits Assigned	03		01		--		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	25	150

Course prerequisites:

Data structure and Discrete Structures.

Course Objectives:

The objective of the course is to teach various techniques for effective problem solving in computing. The different algorithm paradigms for problem solving will be used to illustrate efficient methods to solve problems. The analysis of the algorithm will be demonstrated to show the efficiency of the algorithm. The complexity theory of the problems is introduced to students for further analysis of algorithms.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1: Analyze the asymptotic running time and space complexity of algorithms.

CO2: Describe various algorithm design strategies to solve different problems and analyze complexity.

CO3: Develop string matching techniques

CO4: Describe the classes P, NP, and NP-Complete

Module No.	Details	Hrs.	CO
1	**Introduction to analysis of algorithm** Performance analysis, space and time complexity, Growth of function-Big-Oh; Omega; Theta Notation, Analysis of insertion sort, Introduction to randomized algorithm, Solving Recurrence Problems by Substitution Method, Recursion Tree Method, Masters Method	**06**	**CO1**
2	**Algorithm Design Techniques**		
	Divide and Conquer Technique General method, Finding minimum and maximum algorithm and analysis, Analysis of Merge sort, Analysis of Quick sort, Fast Fourier Transform, Pattern matching using divide and conquer	**06**	**CO2**
	Greedy Technique General method, Knapsack problem, Minimum cost spanning trees-Kruskal's and Prims algorithm, Single source shortest path	**06**	**CO2**
	Dynamic Programming Technique General method, Multistage graphs, 0/1 knapsack, Travelling salesman problem, Single source shortest path, All pairs shortest path, Matrix chain multiplication	**07**	**CO2**
	Backtracking Technique General method, Sum of subsets, 8 queens problem, Graph coloring	**04**	**CO2**
	Branch and Bound General method, 0/1 Knapsack, 15 Puzzle Problem	**04**	**CO2**
3	**String Matching Algorithms** The naïve string-matching Algorithms, String matching with finite automata, The Knuth-Morris-Pratt algorithm, Longest common subsequence	**06**	**CO3**
4	**Non-deterministic Polynomial Algorithms** Polynomial time, Polynomial time verification, NP Completeness and reducibility, NP Completeness proof: Vertex Cover Problem, Clique Problem	**06**	**CO4**
	#Self-Learning Topic- Rod cutting algorithm, randomization algorithms		
TOTAL		**45**	

 # Students should prepare all Self Learning topics on their own. Self-learning topics will enable students to gain extended knowledge of the topic. Assessment of these topics may be included in IA.

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1	T. H. Coreman, C.E. Leiserson, R.L. Rivest, and C. Stein	"*Introduction to Algorithms*",	PHI Publication	2nd Edition, 2005
2	Ellis Horowitz, Sartaj Sahni, S. Rajsekaran,	"*Fundamentals of Computer Algorithms*"	University Press	2nd Edition, 2008
3	Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman	"*Data Structures and Algorithm*"	Pearson education	4th Impression 2009
4	Michael Gooddrich & Roberto Tammassia	"*Algorithm Design Foundation, Analysis and Internet Examples*"	Wiley Student Edition	2nd Edition.

Course Code	Course Title							
116U01C403	Relational Database Management Systems							
	TH		P		TUT		Total	
Teaching Scheme (Hrs./Week)	03		02		--		05	
Credits Assigned	03		01		--		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	-	-	25	150

Course prerequisites:

Data Structure and programming knowledge

Course Objectives:

The objective of the course is to design and program database systems. It covers ER (Entity-Relationship) approach to data modeling, the relational model of Database systems (DBMS) and efficient database design using normalization. It covers Relational Algebra and use of Query Languages such as SQL. This course also introduces Transaction Management, Concurrency Control and Recovery Techniques. The course achieves balance between firm theoretical foundation to designing moderate size databases and creating, querying and implementing realistic databases.

Course Outcomes:

At the end of successful completion of the course the student will be able to

CO1: Understand the features of Relational database management systems.

CO2: Develop relational database design using the designed Entity-Relationship model.

CO3: Use SQL for Relational database creation, maintenance and query processing

CO4: Understand and analyze indexing, hashing, Query processing, query optimization, and Normalization of relational database.

CO5: Apply the transaction, concurrency and recovery techniques

Module No.	Unit No.	Details of Topic	Hrs.	CO
1	**Introduction**		**05**	CO1
	1.1	Introduction, Characteristics of databases, Comparison of File system and Database approach, Users of Database system, Concerns when using an enterprise database		
	1.2	Data Independence, DBMS system architecture, Database Administrator		
2	**Data Modeling: Enhanced-Entity-Relationship Model and Relational Data Model**		**10**	CO2
	2.1	Introduction, Benefits of Data Modeling, Types of Models, Phases of Database Modeling, The Entity-Relationship (ER) Model		
	2.2	Enhanced -Entity-Relationship (EER)- Model Generalization, Specialization and Aggregation		
	2.3	Relational Model:Introduction, Data Manipulation, Data Integrity, Advantages of the Relational Model		
	2.4	Mapping EER Model to Relational Model		
3.	**Relational Algebra and Structured Query Language (SQL),**		**08**	CO2
	3.1	Relational Algebra, Relational Algebra Queries		
	3.2	Overview of SQL, Data Definition Commands, Domain Constraints, Referential integrity		
	3.3	Set operations, aggregate function, null values, Data Manipulation Commands		
	3.4	Data Control commands, Views in SQL, Nested and complex queries, Assertions, Trigger, Security and authorization in SQL		
4	**Query Processing and optimization**		**08**	CO4
	4.1	**Indexing**: Basic concepts, ordered indices: dense and sparse, multilevel indices, secondary indices		
	4.2	**Hashing**: Static hashing, dynamic hashing, comparison of ordered indexing and hashing		
	4.3	**Query processing**: Steps involved in query processing, measures of query cost, algorithms for SELECT and PROJECT operations.		
	4.4	**Optimization**: Overview, Transformation of relational expressions, Estimating statistics, Choice of evaluation plan		

5	**Relational–Database Design**		**07**	CO4
	5.1	First Normal Form, Pitfalls in Relational-Database designs		
	5.2	Function Dependencies, Armstrong Axioms		
	5.3	2nd, 3rd, BCNF and 4th normal form		
	5.4	Decomposition, desirable properties of decomposition		
	5.5	Overall database design process		
6	**Transaction Management, Concurrency control and Recovery protocols**		**07**	CO5
	6.1	Transaction concept, Transaction states, ACID properties		
	6.2	Characterizing schedule based on recoverability and serializability		
	6.3	Concurrency Control: Two-Phase Lock-based ,Timestamp-based, Multi-version Concurrency Control, Validation-based protocols, Deadlock Handling-Wait for graph		
	6.4	Recovery System: Recovery concept, Log based recovery, Shadow paging		
		Total	**45**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1	Elmasri and Navathe	*"Fundamentals of Database Systems"*	Pearson Education	6th Edition
2	Korth, Silberchatz, Sudarshan	*"Database System Concepts"*	McGraw Hill	6th Edition
3	Raghu Ramakrishnan, Johannes Gerhke	*"Database Management Systems"*	McGraw Hill	6th Edition
4	G. K. Gupta	*"Database Management Systems"*	McGraw Hill.	6th Edition

Course Code	Course Title							
116U01C404	Theory of Automata with Compiler Design							
	TH		P		TUT		Total	
Teaching Scheme (Hrs./Week)	03		--		01		04	
Credits Assigned	03		--		01		04	
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	30	20	50	25	--	--	--	125

Course prerequisites:

Students should be familiar with concepts of discrete structures.

Course Objectives

Aims to build concepts regarding the fundamental principles of Grammars, Automata Theory, Turing Machines, PushDown Automata, Un-decidability and Intractable Problems. It aims to understand the design of computing machines that can perform complex computation.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1: Describe regular languages using Regular Expressions, Finite Automata, Nondeterministic Finite Automata, Mealy Machines, Moore Machines and its applications

CO2: Write, simplify and normalize context free grammars and describe context free languages using context free grammar and push down automata

CO3: Design Turing Machines for various problems and its applications

CO4: Understand the concept of Un-decidability and Recursively Enumerable Languages

Module No.	Unit No.	Details	Hrs.	CO
1	**Finite Automata**		**08**	**CO1**
	1.1	Introduction: Alphabets, String, Language, Basic Operations on language, Concatenation, Kleene Star Introduction to different phases of compiler.		
	1.2	Finite. Automata (FA) -its behavior; DFA -Formal definition, state transition diagram, transition table, Language of a DFA. NFA -Formal definition, state transition diagram, transition table Language of an NFA. FA with epsilon-transitions, Eliminating epsilon-transitions, Equivalence of DFAs and NFAs, Conversion from NFA to DFA. Moore machine and Mealy Machine- Formal definition, state transition diagram, transition table, Conversion from Mealy to Moore machine and Moore to Mealy machine. Application of Finite Automata for Lexical Analysis and Lex tools		
2	**Regular Languages**		**09**	**CO1**
	2.1	Chomsky hierarchy, Regular sets, Regular Expression, Some closure properties of Regular languages -Closure under Boolean operations, reversal, homomorphism, inverse homomorphism, etc..		
	2.2	FA and Regular Expressions, equivalence between FA and regular expressions		
	2.3	Pumping lemma for Regular languages, Equivalence and minimization of Finite Automata, Myhill-Nerode Theorem		
	2.4	Application of finite automata and regular expression in lexical analysis		
3	**Context Free Grammars**		**07**	**CO2**
	3.1	Context-free Grammars (CFGs) -Formal definition, sentential forms, leftmost and rightmost derivations, the language of a CFG. Derivation tree or Parse tree-Definition, Simplification of CFGs -Removing useless symbols, epsilon-Productions, and unit productions		
	3.2	Relationship between parse trees and derivations. Parsing and ambiguity, Application of CFGs, Ambiguity in grammars and Languages.		
	3.3	Normal forms -CNF and GNF. Proving that some languages are not context free -Pumping lemma for CFLs, applications. Some closure properties of CFLs -Closure under union, concatenation, Kleene closure, substitution, Inverse homomorphism, reversal, intersection with regular set, etc. Some more decision properties of CFLs		

4	**Push Down Automata**		**07**	**CO2**
	4.1	Pushdown Automata (PDA) -Formal definition, behavior and graphical notation, Instantaneous descriptions (Ids),		
	4.2	The language of PDA (acceptance by final state and empty stack). Equivalence of acceptance by final state and empty stack, Equivalence of PDAs and CFGs,		
	4.3	Conversion: CFG to PDA, PDA to CFG.		
	4.4	DPDAs -Definition, DPDAs, Multi-stack DPDAs & NPDAs and CFLs, Languages of DPDAs, NPDAs		
5	**Turing Machine**		**10**	**CO3**
	5.1	Turing Machines TM -Formal definition and behavior, Transition diagrams, Language of a TM, TM as accepters deciders and generators. TM as a computer of integer functions, Design of TMs, Programming techniques for TMs - Storage in state, multiple tracks, subroutines, etc.		
	5.2	Universal TMs, Variants of TMs –Multi-tape TMs, Nondeterministic TMs. TMs with semi-infinite tapes, Multi-stack machines, Simulating TM by computer, Simulating a Computer by a TM		
	5.3	Equivalence of the various variants with the basic model.		
	5.4	Introduction to parsers, types of parser		
	5.5	Application of CFG, PDA, TM in parsing		
6	**Un-decidability and Recursively Enumerable Languages:**		**04**	**CO4**
	6.1	Recursive and Recursively Enumerable Languages. Properties of Recursive and Recursively Enumerable Languages.		
	6.2	Decidability and Undecidability, Halting Problem, Rice's Theorem, Greibach's Theorem, Post Correspondence Problem, Context Sensitivity and Linear Bound Automata.		
		TOTAL	**45**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman	"Introduction to Automata Theory, Languages and Computation"	Pearson Education	Third Edition, 2006
2.	J.C.Martin,	"Introduction to languages and the Theory of Computation"	TMH	Fourth Edition, 2010
3.	Michael Sipser	"Theory of Computation"	Cengage Learning	Third Edition, 2012
4.	O.G.Kakde	"Theory of Computation"	Laxmi Publications	First Edition, 2007

Term-Work will consist of Tutorials covering the entire syllabus. Students will be graded based on Continuous Assessment of their term work.

Course Code	Course Title						
116U01L401	Web Programming Laboratory						
	TH		P		TUT		Total
Teaching Scheme(Hrs.)	--		04		--		04
Credits Assigned	--		02		--		02
Examination Scheme	Marks						
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	--	--	--	50	--	--	50	100

Course prerequisites:

Basic Programming skills.

Course Objectives

Objective of this course is to provide students an overview of the concepts required for development of application based on Web Technologies.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1. Design dynamic web pages using various HTML tags.

CO2. Use CSS to prepare the layout of web pages.

CO3. Apply JavaScript for validation in client side programming.

CO4. Integrate server side pages using php.

CO5. Apply database operations by integrating SQL queries and session variables.

Module No.	Unit No.	Details	Hrs.	CO
1	**HTML,DHTML**			
	1.1	Designing of effective web site, Introduction to different Web Technologies. HTML Tag Reference, Global Attributes, Event Handlers, Document Structure Tags, Formatting Tags, Text Level formatting, Block Level formatting, List Tags, Hyperlink tags, Image and Image maps,	4	1
	1.2	Table tags, Form Tags, Frame Tags, Executable content tags.		
2	**CSS and Bootstrap**		4	2
	2.1	What are style sheets?, Why are style sheets valuable? Different approaches to style sheets, Using Multiple approaches, Linking to style information in separate file, Setting up style information, Using the <LINK> tag, embedded style information, Using <STYLE> tag, Inline style formation		
	2.2	Introduction to Bootstrap, Bootstrap grids, layouts, bootstrap components like iconography, dropdowns, input groups, navigation, alerts. and plugins		
3	**JavaScript**		6	3
	3.1	Introduction to JavaScript, Data Types, Operators, Control Flow, Arrays, and Functions		
	3.2	Making Decisions / Repeating Code; Debugging and Error Handling; Working with DOM and DHTML		
	3.3	Enhancing and Validating Forms		
4	**PHP Programming**		7	4
	4.2	PHP : Why PHP and MySQL?, Server-side web scripting, Installing PHP, Adding PHP to HTML, Syntax and Variables, Passing information between pages, Strings, Arrays and Array Functions, Numbers, Handling basic PHP errors / problems.		
5	**PHP and MySQL**		9	5
	5.1	PHP/MySQL Functions, Displaying queries in tables, Building Forms from queries, String and Regular Expressions, Sessions, Cookies, Integration of complete web application and deployment.		
	5.2	#self learning topic: Study of Laravel framework		
		Total	**30**	

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	Thomas Powell	Web Design The complete Reference	Tata McGrawHill	5th edition 2010
2.	Thomas Powell	HTML and XHTML The complete Reference	Tata McGrawHill	5th edition 2010
3.	Thomas Powell and Fritz Schneider	JavaScript 2.0 : The Complete Reference,	Tata McGrawHill	3rd Edition,2013
4.	Steven Holzner	PHP : The Complete Reference	Tata McGrawHill	2nd edition 2008

Term-Work consists of programming assignments covering entire syllabus. Students will be graded based on continuous assessment of their term work.

Course Code	Course Title							
116U01P401	Mini Project							
	TH			P		TUT		Total
Teaching Scheme (Hrs.)	01			02		--		03
Credits Assigned	--			03		--		03
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	-	-	-	50	-	-	50	100

Course Objectives:

The objective of Mini Project is to address a real-world problem, find, implement and demonstrate the solution for the same through the courses learned in earlier semesters. Identify various hardware and software requirements for problem solution. It will also inculcate qualities such as meeting deadlines, making and following work plan. The Mini Project may be beyond the scope of courses learnt and interdisciplinary in nature.

Course Outcomes:

After successful completion of the course student should be able to

CO1. Understand the requirements for problem definition and scope.

CO2. Identify the various hardware and software, usage of Tools needed to meet the desired specification.

CO3. Describe the design in the form of algorithm/flowchart/block diagram.

CO4. Implement and test the design as per mentioned specifications.

CO5. Prepare a technical report based on Mini project.

CO6. Present technical seminar based on the Mini project work carried out.

Course delivery guidelines related to Software Engineering principles:

In Theory class teacher is expected to discuss and demonstrate the SDLC process based on waterfall Model.

Module No.	Unit No.	Details	Hrs.	CO
1	**Understand the problem definition (Analysis phase)**		**03**	**CO1**
	1.1	Identify users (different users/stakeholders)		
	1.2	Identify role of every user. who and for what ? (view user, transaction user, admin user etc.)		
	1.3	Is there any parallel system? (survey concept.)		
	1.4	Demonstrate the systems known to students like LMS, MS-Teams, Somaiya web site, with which students are familiar, or the demos prepared/compiled by faculty		
	1.5	Identify need of the project		
	1.6	Prepare problem definition, including background, and scope of the work		
2	**Identification of data and hardware software Requirement (Analysis phase)**		**02**	**CO2**
	2.1	Identify data		
	2.2	Identify software , and hardware needed		
	2.3	Prepare E-R schema or equivalent description of data		
	2.4	Identify usage of Tools, library functions, APIs/packages, applicable to solve the problem		
3	**Identify Design description (Design phase)**		**03**	**CO3**
	3.1	Identify Design description		
	3.2	Represent the definition in the form of block diagram, flowchart, use case diagram		
	3.3	Identify flow of information		
	3.4	Represent information flow in the graphical diagram (web graph, flowchart etc.)		
	3.5	Design user interfaces as per role defined in 1.3 (UI with HTML or equivalent Tool		
	3.6	Describe data in details (e g E-R to Relational Schema)		
	3.7	Define functions as per mentioned scope		
4	**Implementation phase (Coding phase)**		**03**	**CO4**
	4.1	Coding of different modules as defined in the scope		
	4.2	Integration of modules (as per 3.3)		
	4.3	Identify test cases related to different modules.		
	4.4	Prepare test cases		
	4.5	Prepare responsibility chart (Test case writer, Tester) in the form of table		

5	**Testing phase**		**01**	**CO4**
	5.1	Test the modules as per written test case		
	5.2	Complete the test case chart (4.5 - responsibility chart)		
6	**Deployment Phase**		**01**	**CO4**
	6.1	Describe need of deployment		
	6.2	Study of GitHub		
	6.3	Upload the project on GitHub		
7	**Preparation of Report**		**02**	**CO5**
	7.1	Understand the different sections of Report template.		
	7.2	Write report specifying points 1 to 6 in corresponding section		
	7.3	Describe contents of Presentation		
8	**Preparation of Presentation for final demonstration as per report sequence**			**CO5**
		Total	**15**	

Term Work and Practical / Oral:

The mini project is a group project. Interdisciplinary projects are also permitted. Each project will be assigned to one faculty member as a supervisor.

There will be continuous assessment and progress report of the project that needs to be maintained by student(s). The final oral will be a presentation based on a demonstration of the project in front of a committee of examiners.

Course Code	Course Title							
116U01L402	Analysis of Algorithms Laboratory							
	TH		P			TUT		Total
Teaching Scheme(Hrs.)	--		02			--		02
Credits Assigned	--		01			--		01
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	--	--	--	25	--	--	25	50

Term work will consist of experiments based on the syllabus of "Analysis of Algorithm (116U01C402)". Students will be graded based on the continuous assessment of their term work. Practical and Oral examination will based on the syllabus of "Analysis of Algorithms" and laboratory experiments.

Course Code	Course Title							
116U01L403	Relational Database Management Systems Laboratory							
	TH		P			TUT		Total
Teaching Scheme(Hrs.)	--		02			--		02
Credits Assigned	--		01			--		01
Examination Scheme	Marks							
	CA		ESE	TW	O	P	P&O	Total
	ISE	IA						
	--	--	--	25	--	--	25	50

Term work will consist of experiments based on syllabus of 'Relational Database Management System' (116U01C403). Students will be graded based on continuous assessment of their term work. Oral and Practical examination will be based on syllabus of 'Relational Database Management System' and laboratory experiments.