SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Syllabus

Second Year

B. Tech. Computer Engineering

(Programme commenced from 2023-24)

(Department of Computer Engineering)

From
Academic Year 2024-25
(SVU –KJSCE 2.0)

(Approved by BOS dated 22nd April 2024,
Academic Council dated 4th July 2024, Item No. 12.04)

From the Desk of Dean Faculty of Engineering and Technology:

In the era of technological revolution, engineering education must evolve to keep pace with the dynamic demands of industry and society. Our engineering institute is committed to fostering a learning environment that nurtures innovation, creativity, and a profound understanding of engineering principles. The **National Educational Policy 2020 (NEP 2020)** framed by the Government of India recommends a holistic, inclusive, and flexible approach to ensure equitable access to quality education across all levels, promote multidisciplinary research, and impart skill-based education with integration of technology.

Somaiya Vidyavihar, with its esteemed legacy in education, has consistently upheld the values of excellence, inclusivity, and innovation. Applicable for **Somaiya Vidyavihar University (SVU)'s** undergraduate engineering programs, the **SVU Scheme 2023** presented here is aligned with the transformative vision of Somaiya Vidyavihar as well as NEP 2020 to cultivate a holistic, experiential, and interdisciplinary approach to engineering education. The **salient features** of the scheme include:

Professional Core and Elective Courses: The curriculum includes state-of-the-art courses that cover both the fundamentals and emerging trends in respective branches of engineering. With an optimal balance between theoretical knowledge and practical application, core courses provide a strong foundation in essential engineering principles, while elective courses offer flexibility for students to explore and specialize in areas of interest.

Open Elective Courses: Recognizing the importance of interdisciplinary knowledge, the curriculum includes a diverse range of Open Electives categorized into four types: Open Elective Technology (OET), Open Elective Humanities, Open Elective Management (OEHM), and Open Elective Generic (OEG). These courses, offered at institute-level, enable students to expand their knowledge across various disciplines, fostering a versatile skill set and adaptability in an ever-evolving global landscape.

Innovation and Project-based Learning (PBL): The curriculum engages students in innovation and PBL through ideation, mini and major projects right from the first year to the final year of engineering. With diverse projects, collaboration, and field work/community engagement initiatives, students gain a profound understanding of engineering concepts and contribute through innovative solutions to the Sustainable Development Goals (SDGs), societal challenges and advancements.

Learning-by-Doing: The curriculum places emphasis on exposure courses through Skill-Based Learning (SBL) and Activity-Based Learning (ABL), focusing on responsibilities towards society, problem-solving abilities, leadership and teamwork, motivation for life-long learning, etc.

Elements of the Indian Knowledge System: The curriculum incorporates aspects of the Indian Knowledge System that emphasize on drawing insights from ancient wisdom and rich intellectual heritage of India to address modern challenges.

Internships and Research: Enabling students to gain industry insights and enhance their employability, the curriculum integrates flexible internship opportunities in Semester VII or VIII, allowing students to gain hands-on experience in industries, government sectors, NGOs, and MSMEs. Alternatively, they can opt for a specialized research project and courses in Semester VIII. Besides this Semester-long Internship, all the students are required to complete a mandatory 10-week internship over four years, with a maximum of 4 weeks dedicated to socially relevant internships and a minimum of 6 weeks in technical domains.

Learning through MOOCs: The curriculum leverages and promotes Massive Open Online Courses (MOOCs) to offer students flexible and diverse learning opportunities. Complementing on-campus education, students can learn through MOOCs for Open Electives – OET and OEHM during the Pre-final and Final Year, as well as Professional Core courses during their Internship.

Student Exchange Programs: The curriculum also offers student exchange programs that promote global exposure and cross-cultural learning, elevating academic and personal growth. Interested students can participate in the Student Exchange Programs as an alternative to the semester-long internship. Credits from the foreign university where they study will be transferred, providing them with an opportunity to experience different educational systems, cultures, and perspectives.

Minors Courses: Students can expand their academic horizons by pursuing minors in disciplines other than their major, earning additional credits. These minor courses provide an opportunity to acquire multidisciplinary knowledge, significantly enhancing their versatility and adaptability in the professional world.

Honors Courses: For high-achieving students, the SVU 2023 scheme offers Honors courses that delve deeper into specialized topics and gain additional credits for the same. These advanced courses align with high-end industry standards and provide an enriched learning experience, offering multiple opportunities to expand knowledge and expertise in areas of interest.

This forward-thinking SVU 2023 scheme is designed to equip our graduating engineers to emerge as innovative leaders, capable of addressing global challenges and contributing to the advancement of society. Our Boards of Studies, comprising experts in different disciplines, have meticulously designed syllabus for various programs under this SVU 2023 Scheme. We are confident that the joint efforts of the faculty, alumni, students, industry experts, and all the stakeholders will strengthen the academic, research, and entrepreneurial culture of our institution, reinforcing K. J. Somaiya College of Engineering’s position as one of the premier engineering institutions in the nation and a top choice for engineering aspirants.

Dr. S. K. Ukarande
Dean – Faculty of Engineering and Technology
Somaiya Vidyavihar University, Mumbai

Board of Studies in Computer Engineering

Dr. Prasanna J. Shete	Chairman
Dr. Parikshit Mahalle	Academician Member
Mr. Chetan Mistry	Industry Member
Dr. G. Sivakumar	Research Institute Member
Mr. Dhaval Chothani	Alumni Industry Member
Dr. Manish Potey	Faculty Member (Professor)
Dr. Bhakti Palkar	Faculty Member (Associate Professor)
Dr. Archana Gupta	Faculty Member (Assistant Professor)

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:

PO 1: ***Engineering knowledge:*** *Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.*

PO 2: ***Problem analysis****: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.*

PO 3: ***Design/development of solutions:*** *Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.*

PO 4: ***Conduct investigations of complex problems:*** *Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.*

PO 5: ***Modern tool usage:*** *Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.*

PO 6: ***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*

PO 7: ***Multidisciplinary Competence:*** *Recognize/ study/ analyze/ provide solutions to real-life problems of multidisciplinary nature from diverse fields*

PO 8: ***Ethics:*** *Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice*

PO 9: ***Individual and team work:*** *Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings*

PO 10: ***Communication:*** *Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.*

PO 11: ***Project management and finance:*** *Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.*

PO 12: ***Life-long learning:*** *Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.*

Program Specific Outcomes (PSO)

PSO 1:	Design, construct and implement hardware and software based modern Computing / Information systems with varying complexities
PSO 2:	Demonstrate competence in designing, implementation and maintenance of computer based applications, computer-controlled equipment and networks of intelligent devices

Acronyms used:

1. Acronyms for Category of course and Syllabus Document

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	**LAB /TUT CA**	Continuous Assessment of Laboratory/ Tutorial
PE	Professional Elective courses	**TH**	Theory
OET	Open Elective - Technical	**TUT**	Tutorial
OEHM	Open Elective - Humanities and Management	**ISE**	In- Semester Examination
LC	Laboratory Courses	**CO**	Course Outcome
PR	Project	**PO**	Program Outcome
EX	Exposure Course	**PSO**	Program specific Outcome

2. 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
W	Workshop
V	Value Based Course

3. Eight Digit Course code e.g. 216U06C301

Acronym Serially as per code	Definition
2	SVU 2023 Second revision
16	College code
U	Alphabet code for type of programme
06	Programme code
C	Type of course
3	Semester number - Semester III
01	Course Number

Semester III

Credit and Examination Scheme

SEMESTER III (With Effect from 2024-25)

Credit Scheme

Course Code	Name of the Course	Teaching Scheme TH-PR-TUT	Total (hrs.)	Credit Scheme TH-PR-TUT	Total Credits	Course Category
216U01C301	Integral Transforms and Vector Calculus	3 – 0 – 1	04	3 – 0 – 1	04	BS
216U01C302	Data Structures	3 – 0 – 0	03	3 – 0– 0	03	PC
216U01C303	Object Oriented Programming Methodology	3 – 0– 0	03	3 – 0 – 0	03	PC
216U01C304	Computer Organization & Architecture	3 – 0--0	03	3 – 0 – 0	03	PC
216U01C305	Discrete Mathematics	3 – 0– 1	04	3 – 0– 1	04	PC
216U06I306	Indian Knowledge System	2 – 0– 0	02	2 – 0– 0	02	IKS
216U01L301	Digital Design Laboratory	0 – 2 – 1	03	0 – 1 – 1	02	LC
216U01L302	Data Structures Laboratory	0 – 2 – 0	02	0 – 1 – 0	01	LC
216U01L303	Object Oriented Programming Methodology Laboratory	0 – 2 – 0	02	0 – 1 – 0	01	LC
216U01L304	Computer Organization & Architecture Laboratory	0 – 2 – 0	02	0 – 1 – 0	01	LC
	Total	**17 – 08– 03**	**28**	**17 – 04 – 03**	**24**	**--**

- To be conducted in class

Examination Scheme

Course Code	Name of the Course	LAB/ TUT CA	CA		ESE	Total	Course Category
			IA	ISE			
216U01C301	Integral Transforms and Vector Calculus	25	20	30	50	125	
216U01C302	Data Structures	--	20	30	50	100	BS
216U01C303	Object Oriented Programming Methodology	--	20	30	50	100	PC
216U01C304	Computer Organization & Architecture	--	20	30	50	100	PC
216U01C305	Discrete Mathematics	50	20	30	50	150	PC
216U06I301	Indian Knowledge System	--	50	--	50	100	PC
216U01L301	Digital Design Laboratory	50	--	--	--	50	IKS
216U01L302	Data Structures Laboratory	50	--	--	--	50	LC
216U01L303	Object Oriented Programming Methodology Laboratory	50	--	--	--	50	LC
216U01L304	Computer Organization & Architecture Laboratory	50	--	--	--	50	LC
	Total	**275**	**150**	**150**	**300**	**875**	**--**

Semester IV

Credit and Examination Scheme

Semester IV

Credit Scheme

Course Code	Name of the Course	Teaching Scheme TH-PR-TUT	Total (hrs.)	Credit Scheme TH-PR-TUT	Total Credits	Course Category
216U01C401	Probability, Statistics and Optimization Techniques**$**	3 -0 -1	04	3 – 0 – 1	04	BS
216U01C402	Analysis of Algorithms	3 - 0 - 0	03	3 – 0 – 0	03	PC
216U01C403	Relational Database Management Systems	3 - 0 - 0	03	3 – 0 – 0	03	PC
216U01C404	Operating Systems	3 - 0 - 0	03	3 – 0 – 0	03	PC
216U06R406	Open Elective (Generic)	3 - 0 - 0	03	3 – 0 – 0	03	OE
216U01L401	Competitive Programming Laboratory	0 – 2 – 1	03	0 – 1 – 1	02	LC
216U01L402	Analysis of Algorithms Laboratory	0 – 2 – 0	02	0 – 1 – 0	01	LC
216U01L403	Relational Database Management Systems Laboratory	0 – 2 – 0	02	0 – 1 – 0	01	LC
216U01L404	Operating Systems Laboratory	0 – 2 – 0	02	0 – 1 – 0	01	LC
216U01L405	Web Programming Laboratory	0 – 2 – 2	04	0 – 1 – 2	03	LC
	Total	**15–10–04**	**29**	**15–05–04**	**24**	**--**

$- Common with IT Branch

Examination Scheme

Course Code	Name of the Course	LAB/ TUT CA	CA		ESE	Total	Course Category
			IA	ISE			
216U01C401	Probability, Statistics and Optimization Techniques	25	20	30	50	125	BS
216U01C402	Analysis of Algorithms	--	20	30	50	100	PC
216U01C403	Relational Database Management Systems	--	20	30	50	100	PC
216U01C404	Operating Systems	--	20	30	50	100	PC
216U06R406	Open Elective (Generic)	--	100			100	PC
216U01L401	Competitive Programming Laboratory	50	--	--	--	50	LC
216U01L402	Analysis of Algorithms Laboratory	50	--	--	--	50	LC
216U01L403	Relational Database Management Systems Laboratory	50	--	--	--	50	LC
216U01L404	Operating Systems Laboratory	50	--	--	--	50	LC
216U01L405	Web Programming Laboratory	75	--	--	--	75	LC
	Total	**300**	**180**	**120**	**200**	**800**	**--**

$- Common with IT Branch

Course Code	Name of the Course
216U01C301	Integral Transform and Vector Calculus

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	--	01	04
Credits Assigned	03	--	01	04

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	25	30	20	50	125

Course pre-requisites:

- 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 (CO):

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

CO 1	Apply Different methods to find Laplace Transform and Inverse Laplace Transform of a function
CO 2	Find Fourier series, Fourier Integral and Fourier Transform of functions.
CO 3	Apply Different methods to find Z-Transform and Inverse Z- Transform of a function.
CO 4	Apply concepts of Gradient, curl and Divergence of a vector function to solve problems.
CO 5	Apply concepts of Vector Integration to solve related problems.

Detailed Curriculum

Module No.	Unit No.	Contents	No of 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**	**CO 2**
	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**		**04**	**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**		**08**	**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**		**09**	**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**	

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/Year
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, Manish Goyal	*A Textbook of Engineering Mathematics*	Laxmi Publications LTD, India	9th Edition 2016
4	P. N. Wartikar, J. N. Wartikar	*A text book of Applied Mathematics Vol I & II*	Pune Vidyarthi Gruha, India	6th Edition 2012

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

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

Course Code	Name of the Course
216U01C302	**Data Structures**

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	-	-	03
Credits Assigned	03	-	-	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	-	20	30	50	100

Course pre-requisites:

- 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 (CO):

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

CO 1	Comprehend the different data structures used in problem solving
CO 2	Apply linear and non-linear data structure in application development.
CO 3	Describe concepts of advance data structures like set, map & dictionary.
CO 4	Demonstrate sorting and searching methods.

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Introduction to Data Structures**		05	CO1
	1.1	Defining Data structure, Types of Data Structures, Abstract Data Type (ADT), Static and Dynamic Implementations Introduction to space and time complexity, O notation		
	1.2	Applications of data structures.		
2	**Linear data structures: Linked List, Stack and Queue**		16	CO2
	2.1	Introduction and Representation of Linked List, Linked List v/s Array, Implementation of Linked List, Circular Linked List, Doubly Linked List, Application – Polynomial Representation and Addition, Other additional applications/Case study.		
		#Self-learning - Sparse matrix addition		
	2.2	The Stack as an ADT, Stack operations, Array Representation of Stack, Linked Representation of Stack, Application of stack – Polish Notation, Recursion and other applications/Case study, **Application of stack in conversion and evaluation of postfix and prefix expression.**		
	2.3	The Queue as an ADT, Queue operation, Array Representation of Queue, Linked Representation of Queue, Circular Queue, Priority Queue, and Double ended queue, Application of Queues – Simulation and other applications/Case study, **Application of queue in Josephus's Problem.**		
3	**Non-linear data structures: Tree and Graph**		12	CO2
	3.1	Basic tree terminologies, Types of trees, Binary tree representation, Binary tree operation, Binary tree traversal, Binary search tree implementation, Threaded binary trees. Different Search Trees -AVL tree, Overview-Trie, Suffix tree, Applications/Case study of trees.		
		#Self-learning Learning – Red-Black and Splay Trees, Multiway Search Tree, #B Tree, #B+ Tree (# Also covered in DBMS in sem IV)		
	3.2	Introduction to graph as a data structure, Terminologies, Representation, Traversals – Depth First Search (DFS) and Breadth First Search (BFS). Applications/Case study of Graphs.		
4	**Set, Map and Dictionary**		07	CO3
	4.1	Set ADT, Set Implementation, and Partitions with Union-Find operations, Tree based partition implementation.		
	4.2	Map ADT, Implementation, Hash Tables Application of Maps		
#Self-earning - *Dictionary ADT, Implementation, Application of Dictionaries, Exploring case studies on use of set, map and dictionary				

Module No.	Unit No.	Contents	No of Hrs.	CO
(*Covered in python programming course in sem I)				
5	**Searching and Sorting**		05	CO4
	5.1	Sort Concept: Sort Stability, in place sort, number of passes, Bubble Sort, Shell sort		
		#Self-learning - Bucket and Radix sort		
	5.2	Search concept: Linear Search, Binary Search, Hashing, collision resolution: Separate chaining, Linear probing, quadratic probing, double hashing		
		Total	45	--

# Self-learning topics will be evaluated through IA and/or Lab.

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/ Year
1	Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed	*Fundamentals Of Data Structures In C*	University Press	Second Edition 2008
2	Michael T Goodrich Roberto Tamassia David Mount	*Data Structure and Algorithm in C++*	Wiley	Second Edition 2011
3	Richard F. Gilberg & Behrouz A. Forouzan	*Data Structures A Pseudocode Approach with C*	CENGAGE Learning	Second Edition 2005
4	Aaron M Tanenbaum Yedidyah Langsam Moshe J Augentstein	*Data structure Using C*	Pearson	Twelfth Impression 2013
5	Jean Paul Tremblay, Paul G. Sorenson	*An introduction to data structures with applications*	Tata McGraw-Hill Education	Second Edition 1984
6	Reema Thareja	*Data structures using C*	Oxford Higher Education	Second edition, 2014

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Course Code	Name of the Course
216U01C303	Object Oriented Programming Methodology

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	--	--	03
Credits Assigned	03	--	--	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	--	20	30	50	100

Course pre-requisites:

- 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, Inheritance, Multithreading and packages.

Course Outcomes (CO):

CO 1	Apply the features of object oriented programming languages. (C++ and Java)
CO 2	Explore classes, objects, arrays, strings in C++ and Java
CO 3	Implement scenarios using collection framework
CO 4	Implement the concepts of interfaces, exceptions, multithreading and packages

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Fundamentals of Object oriented Programming**		10	CO 1
	1.1	Introduction, Basic Program Construction, 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++.		
	1.3	**C++ Programming Basics:** Namespace Fundamentals, using, The standard Namespace, Data types, Input with cin, Output Using cout, Type Conversion: Automatic Conversions, Casts		
	1.4	Loops and Decision making statements, Functions, Function overloading		
2	**Class, Object, Method and Constructor**		08	CO 1, CO 2
	2.1	Class Object and Method: member, method, Modifier, Selector, iterator, State of an object. Memory allocation of object using new operator, Command line Arguments. instanceof operator in Java.		
	2.2	Method overloading & overriding, constructor, destructor in C++, Types of constructor (Default, Parameterized, copy constructor with object), Constructor overloading, this, final. super keyword, Garbage collection in Java.		
3	**Arrays and Strings**		07	CO 2, CO 3
	3.1	Arrays:1D, 2D, Variable Length array, for-each with Array		
	3.2	Array of objects, Vector, ArrayList		
	3.3	Wrapper class in Java		
4	**Inheritance and Polymorphism**		11	CO 1, CO 4
	4.1	Inheritance: Types of Inheritance, inheritance using inversion of control		
	4.2	Final class, abstract class with constructor, abstract and non-abstract methods, Method Overriding.		
	4.2	Interface, final keyword, Implementing interfaces, extending interfaces, Difference between an Abstract class and an Interface		

5	**Packages, Exception Handling and Multithreading in Java**		10	CO4
	5.1	Packages: Creating Packages, Using Packages, Access Protection, Predefined packages		
	5.2	Exception handling: Exception as objects, Exception hierarchy, Try catch finally Throw, throws		
	5.3	Multithreading: Thread life cycle, Multithreading advantages and issues, Simple thread program, Thread synchronization.		
		Total	**45**	--

Reference Books

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/ Year
1	Herbert Schildt	*Java: The Complete Reference*	Tata McGraw-Hill	12th Edition, 2021
2	Kathy Sierra, Bert Bates	*OCA Java SE 8 Programmer I Exam Guide*	McGraw-Hill Edition	6th Edition, 2017
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	Yashwant Kanetkar	*Let us C++*	BPB publications	16th Edition, 2020
6	Ralph Bravaco,Shai Simonson	*Java Programming from the Ground up*	Tata McGraw-Hill	2012

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Course Code	Name of the Course
216U01C304	Computer Organization and Architecture

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	-	-	03
Credits Assigned	03	-	-	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	-	20	30	50	100

Course pre-requisites: 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 (CO):

CO 1	Describe and define the structure of a computer with buses structure and detail working of the arithmetic logic unit and its sub modules
CO 2	Understand the Central processing unit with addressing modes and working of control unit in depth.
CO 3	Learn and evaluate memory organization and cache structure
CO 4	Summarize Input output techniques and multiprocessor configurations

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**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, difference between computer architecture and computer organization.		
	1.2	Introduction to buses, bus types, and connection I/O devices to CPU and memory, PCI and SCSI		
2	**Arithmetic and Logic Unit**		10	CO1
	2.1	Booth's Recoding and Booth's algorithm for signed multiplication, Restoring division and non-restoring division algorithms.		
	2.2	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	**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 pipelining		
		Self learning: RISC and CISC Architecture, Case study on SPARC		
4	**Memory Organization**		10	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, Introduction to: Virtual memory, Main memory allocation, Segmentation ,Paging: demand paging and thrashing. Secondary storage, RAID levels		
5	**I/O Organization**		03	CO4
	5.1	External Devices, I/ O Modules		
	5.2	Programmed I/O, Interrupt driven I/O, DMA		

Module No.	Unit No.	Contents	No of Hrs.	CO
6		**Multiprocessor Configurations**	**08**	**CO4**
	6.1	Flynn’s classification, Parallel processing systems and concepts, Introduction to pipeline processing and pipeline hazards.		
	6.2	Design issues of pipeline architecture, Instruction pipelining: Six Stage instruction pipeline.		
	6.3	8086 Instruction set (Arithmetic Instructions, Logical Instructions, Data transfer instructions), Assembly language programming.		
		Total	**45**	**--**

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/ Year
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	

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Course Code	Course Name
216U01C305	**Discrete Mathematics**

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	-	01	04
Credits Assigned	03	-	01	04

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	25	20	30	50	125

Course pre-requisites:
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 (CO):

CO 1	Use various mathematical notations, apply various proof techniques to solve real world problems
CO 2	Learn and apply core ideas of Set Theory, Relations & Functions
CO 3	Use graphs and their types, to solve the practical examples
CO4	Understand the use of Algebraic structures and lattice, to solve the problems

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Set Theory**		**03**	**CO 1, CO 2**
	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	CO 1
	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**	**CO 2**
	3.1	Relations, Paths and Digraphs		
	3.2	Properties and types of binary relations.		
	3.3	Manipulation of relations, Composition, Closures, Warshall's algorithm		
	3.4	Equivalence relations, equivalence classes.		
4	**Posets and Lattice**		**10**	**CO 2 , CO 4**
	4.1	Partial ordered relations (Posets), Hasse diagram		
	4.2	Lattice, sublattice		
	4.3	Types of Lattice , distributive lattice, complementary Lattice, Boolean Algebra		
5	**Functions and Pigeon Hole Principle**		**03**	**CO 2**
	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**	**CO 3**
	6.1	Definitions, Paths and circuits, Types of Graphs , Eulerian path and circuit and Hamiltonian path and circuit.		
	6.2	Planer graphs		
	6.3	Isomorphism of graphs		
	6.4	Subgraph		
7	**Algebraic Structures**		**12**	**CO 4**
	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		
		Total	**45**	--

Reference Books

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/Year
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

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Course Code	Course Name
216U06I306	**Foundational Course in Indian Knowledge Systems**

	TH	P	TUT	Total
Teaching Scheme (Hrs./Week)	02	-	--	02
Credits Assigned	02	-	--	02

Evaluation Scheme	**Marks**				
	LAB/TUT CA	**CA (TH)**		**ESE**	**Total**
		IA	**ISE**		
	--	50	--	50	100

Course Objectives:

1. To introduce students to the rich diversity of Indian knowledge systems
2. To introduce the life and works of important figures in the respective domains
3. To explore the underlying philosophical and cultural ethos that distinguishes Indian Knowledge Systems
4. To emphasise continuity of the tradition into modern times, wherever applicable.

Course Outcomes

At the end of successful completion of the course the student will

CO 1	Have a clear understanding of the different domains of Indian Knowledge Systems
CO 2	Have become aware of the contribution of great figures in the respective fields
CO 3	Have an understanding of how culture impacts creation of knowledge
CO 4	Learn to investigate correlations and synthesis leading to development of any knowledge system

Module No.	Unit No.	Topics	Hours per topic	Hours per Unit
1	**Sources of Indian Knowledge Systems**			**04**
	1.1	IKS - Concept, scope, relevance to our world today.	1	
	1.2	Textual sources, historical accounts, archaeological evidence, inscriptions, coins etc	3	
2		**Why study IKS?**		**02**
	2.1	Importance of the IKS, its interconnections and relevance to the modern fields of science.	2	
3	**Yoga: Basic Practices and Philosophy**			**06**
	2.1	Maharshi Patanjali, Swami Satyananda Saraswati, B K S Iyengar, Swami Kuvalayananda, Sri Yogendra	2	
	2.2	Body loosening exercises Importance of breath, developing concentration Yoga for mind-body wellness	4	
4	**Genres of Ancient Literature**			**06**
	4.1	Religious: Vedic texts, Buddhist and Jain texts;	2	
	4.2	Epics, Puranas, Sangam literature	2	
	4.3	Poetry, Mathematics, and Scientific Literature	2	
5		**Leadership and Ethical Values**		**06**
	5.1	Selections from Shantiparva of Mahabharata, Arthasastra, Panchatantra, Hitopadesha, Jataka tales, Bhagavadgita, Dhammapada and Thirukkural: Discussions on ethical values	3	
	5.2	Leadership qualities as reflected through ancient Indian literature: Lessons for modern leadership challenges	3	
6			3	
	6.1 Art: Sculpture(iconography) and Paintings			
		Iconography: Ellora (Buddhist and Jain) and Hampi (Hindu)		
		Paintings: Ajanta(Buddhist), Ellora(Jain), Brihadeshvar Temple-Thanjavur.(Hindu)		
	6.2 **Architecture: Rock-cut caves and Temple Architecture**		3	
		Rock-cut caves: Kanheri, Elephanta, Ellora (any two sites can be used for detailed discussion)		
		Temple architecture: Pattadakal, Konark Temple, Jagannatha Temple-Puri, Bodh Gaya, Dilwara Temple-Mount Abu (any two sites can be used for detailed discussion)		
7	**Ancient Indian Mathematics**			**06**
	7.1	Shulba Sutras, Bakshali Manuscript	2	

Module No.	Unit No.	Topics	Hours per topic	Hours per Unit
	7.2	Aryabhatiya: place value system, approximation of the value of π, geometry	2	
	7.3	Bhaskaracharya: different approach to teaching mathematics	2	
8	**Ancient Indian Astronomy**			**06**
	8.1	Indian calendar system: Sayana-nirayana calendar, Panchanga	3	
	8.2	Spherical trigonometry, Eclipse computation	3	
9	**Ancient Indian Agriculture**			**06**
	9.1	General management of Agriculture and Farming Operations	3	
	9.2	Cattle Management, Weather predictions	3	
10	**Trade and Commerce**			**06**
	10.1	Silk route, Uttarapatha and Dakshinapatha, Maritime route	3	
	10.2	Barter system, Numismatics	3	
11	**Ancient Indian Society**			**06**
	11.1	Law and Justice	3	
	11.2	Marriage Laws, Inheritance	3	
12	**Chemistry and Metallurgy**			**06**
	12.1	Multiple sources such as archaeological artifacts, temple icons,	1	
	12.2	Metals and beads	2	
	12.3	Chemistry of dyes, Colouring materials	2	
	12.4	Paintings and Painting materials	1	
Total Hours				**30***

* The first two modules remain the core and other modules can be selected (any 4 modules from module 3 to module 12) by the college depending upon the availability of the teachers making it to a 30hrs course.

Recommended Books:

Text Book on IKS:

1. Mahadevan B., Bhat Vinayak Rajat, Nagendra Pavana R. N. Introduction to Indian Knowledge System: concepts and Applications, PHI Learning Pvt. Ltd. 2022

Reference Books:

1. Amma Sarasvati T. A., Geometry in Ancient and Medieval India, MLBD, Delhi, 1st ed. 1999, reprint 2007
2. Acharya, P. K., Indian Architecture According to ManasaraShilapshastra, Oxford University Press 1927
3. Altekar, A.S., Education in Ancient India, Gyan Books, 2010
4. Appleton Naomi, Jataka Stories in Theravada Buddhism: Narrating the Bodhisatta Path, Routledge Publication, New York 2016
5. Bhatacharyya, T. , Study of Vastuvidya or Canon of Indian Architecture, Patna 1976
6. Bose, N. K., Orissan temple Temple Architecture (Vastushastra) [With Sanskrit text and English translation), Bharatiya Kala Prakashana, Delhi 20017
7. Chatterjee, Satischandra & Datta, Dhirendra Mohan. An introduction to Indian Philosophy, Rupa Publications India Pvt. Ltd., New Delhi, 7th edition, 1968
8. Clark Walter Eugene, The Aryabhatiya of Aryabhata- An Ancient Indian Work On Mathematics and Astronomy, Delta Book World, India, 2021
9. Coomaraswamy, Ananda K. Early Indian Architecture: Cities and City-Gates, Munshiram Manoharlal Publishers, 2002
10. D M Bose, S N Sen and B V Subbarayappa, eds; A Concise History of Science in India, INSA; 2009
11. Datta Bibhutibhushan & Singh Avadhesh Narayan, History of Hindu Mathematics, 1935, repr. Bharatiya Kala Prakashan, Delhi, 2004
12. Datta Bibhutibhushan, Ancient Hindu Geometry: The Science of the Śulba, 1932, reprint. Cosmo Publications, New Delhi, 1993
13. Deglurkar, G. B, Temple Architecture and Sculpture of Maharashtra, Nagpur University, Nagpur 1974
14. Dehejia, Vidya, Early Buddhist Rock Temples A Chronological Study, London, 1972
15. Dehejia, Vidya, Early Stone Temples of Orissa, Vikas Publishing House, Delhi 1979
16. Divakaran P. P., The Mathematics of India: Concepts, Methods, Connections, Hindustan Book Agency, 2018
17. Dr. Mishra Shiv Shekhar, Fine Arts & Technical Sciences in Ancient India with special reference to Someśvara's Mānasollāsa; Krishnadas Academy, Varanasi 1982
18. Ed. and Trs. Majumdar Girija Prasanna, Banerji Sures Chandra, Krisi-Parasara, Asiatic Society, Kolkata, 1960
19. Ed. Tr. Kangale, R. P, Kautiliya Arthashastra, University of Bombay, Bombay, 1960
20. Gupta, Swarajya Prakash, Asthana Shashi, Elements of Indian Art: Including Temple Architecture, Iconography & Iconometry, Indraprastha Museum of Art and Archeology, 2007
21. Kane P.V., History of Sanskrit Poetics, Motilal Banarasidass, New Delhi, 4th edition, 1971

22. Larson, G. J. (Ed.) and Bhattacharya, R. (Ed.) , Encyclopaedia of Indian Philosophies: Yoga: India’s Philosophy of Meditation, Vol. XII, Motilal Banarasidas Publishers Pvt. Ltd., Delhi, 1st edi., 2008
23. Paranjpe Kalpana, Ancient Indian insights and Modern Science: A Rare Book, Bhandarkar Oriental Research Institute, Pune, 2022
24. Radhakrishnan, S., The Principal Upanisads, Oxford University Press, Delhi, 1992
25. Rahman A., Alvi M. AKhan .S A., Ghori, Murthy Samba K. V., Science and Technology in Medieval India - A Bibliography of Source Materials in Sanskrit, Arabic and Persian, 1982
26. Rao Balachandra S., Indian Astronomy – An Introduction, Universities Press (India) Limited, Hyderabad, 2000
27. Rao Balachandra S., Indian Mathematics and Astronomy: Some Landmarks, Jnana Deep Publications, Bangalore, 3rd edn, 2004
28. Rao, S. Balachandra, Ancient Indian Astronomy, Planetary Positions and Eclipses, B.R. Publications, 2000
29. Satwalekar S.D., Mahabharata, Svadhyay Mandal, paradi, 1968
30. Sharma Sharmishtha, Buddhist Avadanas, (Socio political, Economic and Cultural Study), Eastern book Linkers, Delhi, 1985
31. Subbarayappa B.V., Science in India: A Historical Perspective, Rupa, New Delhi, 2013
32. Taimini, I. K. , The Science of Yoga, The Philosophical Publishing House, Adyar, 1999
33. Vālmīkīyarāmāyaṇa, Nag Publishers, Delhi, 1990
34. Vatasyayan, Kapila. The Square and the Circle of the Indian Arts, Abhinav Publication, 1997.

Notes to the teachers:

1) Pedagogy:

For effective content delivery, innovative teaching-learning methods will be extremely important. The use of ICT tools, resources from museums and other websites, can be tapped and a repository of common resources can be created across institutions. Classroom sessions can be supplemented with site-based learning at a heritage site nearby, a local museum or a shrine, when studying ancient Indian art and architecture. Immersive experiences will have a powerful impact in units on Yoga, and the performing arts.

The teacher should also make all efforts to incorporate analysis from multiple dimensions such as history, cultural and local significance, and contribution to the economic, social, cultural or literary developments.

The multidisciplinary dimension of each unit should be explored and all attempts must be made to sensitise students to the syncretic nature of Indian culture.

2) Units in the Syllabus:

Recognising that it may not always be possible to find trained faculty for some of the units a list of alternative units is provided in the above syllabus. The first two modules remain the core and other modules can be selected (any 4 modules from module 3 to module 12) by the college depending upon the availability of the teachers making it to a 30hrs course.

3) Assessment:

The model syllabus recommends a 50:50 ratio between Continuous Assessment and End Semester Examination. Innovative methods like MCQ test on each unit, report on a site visit in written or video format, role play and group discussions, or an essay on the experiential content of Yoga, could be included in continuous assessment. The End Semester Exam can be in open book format.

Course Code	Name of the Course
216U01L301	Digital Design Laboratory

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	--	02	01	03
Credits Assigned	--	01	01	02

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	50	--	--	--	50

Course prerequisites: N/A

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, and counters K-maps. The course also includes use of VHDL in the design, simulation, and testing of digital circuits.

Course Outcomes (CO):

At the end of the course students 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.

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Binary Arithmetic and Codes:**		**2**	**CO1**
	1.1	Introduction to Arithmetic and Logical unit, Computer Arithmetic: Fixed- and Floating-point numbers, Signed numbers, Integer Arithmetic,2's Complement arithmetic.		
	1.2	Binary Addition and Subtraction (1's and 2's complement method)		
	1.3	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)		
	3.2	Multiplexers and Demultiplexers, Decoders,		
	3.3	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**	--

F

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/Year
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	NA

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Course Code	Name of the Course			
216U01L302	**Data Structures Laboratory**			
Teaching Scheme (Hrs./Week)	**TH**	**P**	**TUT**	**Total**
	--	02	--	02
Credits Assigned	--	01	--	01

Evaluation Scheme	**Marks**				
	LAB/TUT CA	**CA (TH)**		**ESE**	**Total**
		IA	**ISE**		
	50	--	--	--	50

Course pre-requisites:

- 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 (CO):

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

CO 1	Comprehend the different data structures used in problem solving
CO 2	Apply linear and non-linear data structure in application development.
CO 3	Describe concepts of advance data structures like set, map & dictionary.
CO 4	Demonstrate sorting and searching methods.

Laboratory experiments covering entire syllabus of the course 216U01C302, ‘Data Structures’. Students will be graded based on continuous assessment during laboratory.

Course Code	Name of the Course
216U01L303	**Object Oriented Programming Methodology Laboratory**

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	--	02	-	02
Credits Assigned	--	01	-	01

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	50	--	--	--	50

Course pre-requisites:

- **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, Inheritance, Multithreading and packages.

Course Outcomes (CO):

CO 1	Apply the features of object oriented programming languages. (C++ and Java)
CO 2	Explore classes, objects, arrays, strings in C++ and Java
CO 3	Implement scenarios using collection framework
CO 4	Implement the concepts of interfaces, exceptions, multithreading and packages

Laboratory experiments covering entire syllabus of the course 216U01C303, ‘Object Oriented Programming Methodology’. Students will be graded based on continuous assessment during laboratory.

Course Code	Name of the Course
216U01L304	**Computer Organization and Architecture Laboratory**

	TH	P	TUT	Total
Teaching Scheme (Hrs./Week)	--	02	-	02
Credits Assigned	--	01	-	01

Evaluation Scheme	**Marks**				
	LAB/TUT CA	**CA (TH)**		**ESE**	**Total**
		IA	**ISE**		
	50	--	--	--	50

Course pre-requisites: Basic concepts of computers and their applications.

Course Objectives: Students will try to:

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

Course Outcomes (CO):

CO 1	Describe and define the structure of a computer with buses structure and detail working of the arithmetic logic unit and its sub modules
CO 2	Understand the Central processing unit with addressing modes and working of control unit in depth.
CO 3	Learn and evaluate memory organization and cache structure
CO 4	Summarize Input output techniques and multiprocessor configurations

Laboratory experiments covering entire syllabus of the course 216U01C304, ‘Computer Organization and Architecture’. Students will be graded based on continuous assessment during laboratory.

Semester IV

Course Code	Name of the Course
216U01C401	**Probability, Statistics and Optimization Techniques**

Teaching Scheme (Hrs./Week)	**TH**	**P**	**TUT**	**Total**
	03	-	01	04
Credits Assigned	03	-	01	04

Evaluation Scheme	**Marks**				
	LAB/TUT CA	**CA (TH)**		**ESE**	**Total**
		IA	**ISE**		
	25	20	30	50	125

Course pre-requisites:

- 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 (CO):

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

CO 1	Apply concepts of Binomial, Poisson, Uniform, Exponential and Normal distribution to solve Engineering problems
CO 2	Apply concepts of correlation and regression for given bivariate and multivariate data.
CO 3	Estimate population parameters and apply large and sample test to analyse collected data.
CO 4	Apply concepts of Linear and Nonlinear programming methods to solve problems
CO 5	Solve problems based on single server limited queue and single server unlimited queue models.

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Probability and Probability Distribution**		12	CO 1
	1.1	Review: Probability, Conditional Probability, Bayes' theorem, Joint Probability		
	1.2	Discrete and Continuous Probability Distribution, Expected value and Variance of Random variables		
	1.3	Skewness and Kurtosis, Quantiles		
	1.4	Binomial Distribution, Poisson Distribution		
	1.5	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		
#Self-Learning topic: Correlation and regression in Multivariate.				
3	**Estimation and Sampling Theory**		12	CO 3
	3.1	Estimation of Parameters: Central Limit Theorem, Unbiased Estimators, Efficiency of Estimators, Estimation of Confidence interval for Mean with known Variance and unknown Variance		
	3.2	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. p-value		
	3.3	Difference between sample mean and population means for large samples, Test for significance of the difference between the means of two large samples.		
	3.4	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.		
	3.5	Chi-square distribution as a Test of Independence, Test of the Goodness of fit and Yate's correction.		
4	**Optimization Techniques**		10	CO4
	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	Unconstrained optimization, problems of two or three variables with one equality constraint using Lagrange's Multiplier method.		
	4.4	Problems of two or three variables with one inequality		

Module No.	Unit No.	Contents	No of Hrs.	CO
		constraint using Kuhn-Tucker conditions		
	4.5	Problems of two or three variables with one inequality constraint using Kuhn-Tucker conditions		
5	**Queuing Theory**		**05**	**CO5**
	5.1	Introduction, Features of Queuing, solution of Queuing models. M/M/1(Single Server, Unlimited Queue Model)		
	5.2	M/M/1 Single Server, limited Queue Model		
		Total	**45**	**--**

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/ Year
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	Richard A. Johnson	*Probability and Statistics for Engineers*	PHI Learning Private Limited	8th Edition 2011
5	Ronald E. Walipole, Raymond H.Myers	*Probabilities & Statistics for Engineers & Scientists*	Pearson Education	9th Edition 2010

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Tutorials will be covering entire syllabus of “Probability, Statistics and Optimization Techniques” (216U01C401). 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.

Course Code	Name of the Course
216U01C402	Analysis of Algorithms

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	--	--	03
Credits Assigned	03	--	--	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	--	20	30	50	100

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 (CO):

CO 1	Analyse the asymptotic running time and space complexity of algorithms.
CO 2	Describe various algorithm design strategies to solve different problems and analyse complexity.
CO 3	Develop string matching techniques.
CO 4	Describe the classes P, NP, and NP-Complete.

Detailed Curriculum

Module No.	Unit No.	Contents	No. of Hrs.	CO
1	**Introduction to analysis of algorithm**		06	CO 1
	1.1	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.		
2	**Algorithm Design Techniques**			
	2.1	**Divide and Conquer Technique** General method, Finding minimum and maximum algorithm and analysis, Analysis of Merge sort, Quick sort and Heap Sort, Strassen's Matrix Multiplication.	06	CO 2
	2.2	**Greedy Technique** General method, Knapsack problem, Job Scheduling with deadlines, Minimum cost spanning trees-Kruskal's and Prims algorithm, Single source shortest path.	06	CO 2
	2.3	**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	CO 2
	2.4	**Dynamic Programming Technique** General method, Multistage graphs, 0/1 knapsack, Travelling salesman problem, Single source shortest path, All pairs shortest path, Matrix chain multiplication.	04	CO 2
	2.5	**Branch and Bound** General method, 0/1 Knapsack, 15 Puzzle Problem.	04	CO 2
3	**String Matching Algorithms**		06	CO 3
	3.1	The naïve string-matching Algorithms, The Knuth-Morris-Pratt algorithm, Longest common subsequence.		
4	**Non-deterministic Polynomial Algorithms**		06	CO 4
	4.1	Polynomial time, Polynomial time verification, NP Completeness and reducibility, NP Completeness proof: Vertex Cover Problem.		
#Self-Learning Topic- Rod cutting algorithm, randomization algorithms				
		Total	**45**	--

Reference Booksits

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/Year
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	Name of the Course
216U01C403	**Relational Database Management Systems**

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	-	-	03
Credits Assigned	03	-	-	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	-	20	30	50	100

Course pre-requisites: Data Structure and programming knowledge.

Course Objectives:

The aim of this course is to equip the students with the knowledge and skills to design and program effective database systems. This includes understanding the Entity-Relationship (ER) approach to data modelling, the relational model of Database Management Systems (DBMS), and efficient database design through normalization. It introduces the relational algebra and query languages such as SQL to retrieve, manipulate, and manage data within a database. Further the concepts of Transaction Management, Concurrency Control and Recovery Techniques in databases striking a balance between providing a strong theoretical foundation for designing databases and practical skills for creating, querying, and implementing realistic databases.

Course Outcomes (CO):

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

CO 1	Comprehend the Characteristics of Relational Database Management Systems.
CO 2	Create Relational Database Designs Based on Entity-Relationship Models.
CO 3	Utilize SQL for Relational Database Operations.
CO 4	Analyse Advanced Database Concepts like indexing, hashing, query processing, query optimization, normalization.
CO 5	Apply Transaction Management, Concurrency Control, and Recovery Techniques

.

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Introduction to Databases**		**05**	**CO 1**
	1.1	Database and Database Users: Introduction, Characteristics of the Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of Using the DBMS approach.		
	1.2	Database System Concepts and Architecture: Data Models, Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, Database System Environment, Centralized and Client/server Architectures for DBMSs, Classification of Database Management Systems.		
2	**Conceptual Data Modeling , Database Design Relational Data Model and Database constraints**		**10**	**CO 2**
	2.1	Using High-Level Conceptual Data Models for Database Design, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, ER Diagrams, Relationship Types of Degree Higher than Two.		
	2.2	The Enhanced Entity- Relationship (EER) Model: Subclasses, Superclasses, and Inheritance, Specialization and Generalization, Constraints and characteristics of Specialization and Generalization Hierarchies, Modeling of UNION types using Categories.		
	2.3	Relational Data Model and Relational database constraints: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas		
	2.4	Relational Database Design by ER and EER-to-Relational Mapping		
3	**Relational Algebra and Structured Query Language (SQL)**		**08**	**CO 3**
	3.1	Relational Algebra: Unary Relational operations: SELECT and PROJECT, Relational algebra operations from Set Theory, Binary Relational Operations: JOIN and DIVISION, Additional Relational Operations.		
	3.2	**SQL:** SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT, DELETE and UPDATE statements in SQL. More Complex SQL retrieval Queries, Specifying constraints as Assertions and Actions as Triggers, Views in SQL		
#Self Learning Security and Authorization using SQL				

Module No.	Unit No.	Contents	No of Hrs.	CO
4	**Relational Database Design, Indexing, Query Processing and Optimization**		14	CO 4
	4.1	**Relational Database Design**: Basic of Functional Dependencies and Normalization for Relational Databases, Inference rules, Equivalence and Minimal Cover, Properties of Relational Decompositions		
	4.2	**Indexing Structure for files and Physical Database Design:** Types of Single-Level ordered indexes, Multi-level indexes, Dynamic Multi-level indexes using B-Trees and B+ Trees		
	4.3	**Query Processing**: Algorithms for SELECT, PROJECT operations		
	4.4	**Query Optimization**: Query Trees and Heuristics for Query optimization		
#Self-learning – Database performance tuning				
5	**Transaction Processing , Concurrency Control and Recovery**		08	CO 5
	5.1	**Transaction Processing concepts**: Introduction to Transaction processing, Transaction and system concepts, Desirable properties of transactions, Characterizing schedules based on recoverability, Characterizing schedules based on serializability		
	5.2	**Concurrency control techniques**: Two-Phase Lock-based ,Timestamp-based, Multi-version Concurrency Control, Validation-based protocols, Deadlock Handling-Wait for graph		
	5.3	**Recovery Techniques**: Recovery concepts, NO-UNDO/REDO Recovery based on deferred update, Recovery techniques based on immediate update, Shadow paging		
		Total	**45**	--

Reference Books

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/Year
1	Elmasri and Navathe	*“Fundamentals of Database Systems”*	Pearson education	7th 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	Name of the Course
216U01C404	Operating Systems

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	03	-	-	03
Credits Assigned	03	-	-	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	--	20	30	50	100

Course pre-requisites: Basics of Computer Organization and architecture.

Course Objectives:

- To introduce basic concepts and functions of operating systems.
- To understand the concept of process, thread and resource management.
- To understand the concepts of process synchronization and deadlock
- To understand various Memory, I/O and File management techniques.
- To understand the designing and implementation of system software like Assembler.
- Macro pre-processor and linker loader

.

Course Outcomes (CO):

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

CO 1	Identify the different system programs and their utility and Explain the fundamental concepts of operating system with extension to Unix and Mobile OS.
CO 2	Illustrate and analyze the Process, threads, process scheduling and thread scheduling.
CO 3	Illustrate and analyze the Process, threads, process scheduling and thread scheduling
CO 4	Explain disk organization and file system structure with illustration of disk scheduling algorithms
CO 5	Understand Storage management with allocation, segmentation & virtual memory concepts

.

Detailed Curriculum

Module No.	Unit No.	Details	No. of Hrs.	CO
1	**Introduction to System software**			
	1.1	Concept, introduction to various system programs such as assemblers, loaders, linkers, macro processors, compilers, interpreters, operating systems, device drivers Operating System Objectives and Functions,	09	CO 1
	1.2	The Evolution of Operating Systems		
	1.3	Operating system structures in detail		
	1.4	System Calls		
	1.5	Linux Kernel and Shell		
	1.6	System boot		
#Self Learning : WINDOWS Booting Process				
2	**Process Concept and scheduling**			
	2..1	Process: Concept of a Process, Process States, Process Description, Process Control Block, Operations on Processes. Threads: Definition and Types, Concept of Multithreading	10	CO 2
	2.2	Multicore processors and threads. Scheduling: Uniprocessor Scheduling - Types of Scheduling: Preemptive and, Non-preemptive, Scheduling Algorithms: FCFS, SJF, SRTN, Priority based, Round Robin, Multilevel Queue scheduling.		
	2.3	Multi Processor Scheduling		
	2.4	Introduction to Thread Scheduling Linux Scheduling.		
# Self Learning : OS Design Considerations for Multiprocessor and Multicore architectures				
3	**Process Concurrency**			
	3.1	Concurrency: Principles of Concurrency, InterProcess Communication, Process/Thread Synchronization.	12	CO 3
	3.2	Mutual Exclusion: Requirements, Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors)		
	3.3	Classical synchronization problems: Readers/Writers Problem, Producer and Consumer problem.		
	3.4	Principles of Deadlock: Conditions and Resource Allocation Graphs, Deadlock Prevention, Deadlock Avoidance: Banker‟s Algorithm for Single & Multiple Resources, Deadlock Detection and Recovery. Dining Philosophers Problem		
4	**Input output and file management**			
	4.1	File Management: Overview, File Organization and Access, File Directories, File Sharing, Secondary Storage Management, Linux Virtual File System, Inode Structure	09	CO 4
	4.2	I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling algorithm: FCFS, SSTF, SCAN, CSCAN, LOOK, CLOOK. Disk Management, Linux I/O.		

Module No.	Unit No.	Details	No. of Hrs.	CO
5	**Storage management**			
	5.1	Main Memory: Background, Swapping, Contiguous Memory Allocation, 32 and 64 bit architecture Examples ,Buffering	05	CO 5
#Self-Learning Component: Android OS, Cloud OS, Contemporary issues and solutions in Memory Management				
		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

Reference Books

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/Year
1	William Stallings	*Operating System: Internals and Design Principles*	Prentice Hall	8th Edition, 2014
2	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne	*Operating System Concepts*	John Wiley & Sons , Inc.	9th Edition, 2016
3	Andrew Tannenbaum	*Operating System Design and Implementation*	Pearson	3rd Edition
4	D.M Dhamdhere	*Systems programming*	Tata Mc-Graw Hill	2nd Edition
5	Maurice J. Bach	*Design of UNIX Operating System*	PHI	2nd Edition
6	J.J Donovan	*Systems Programming*	Tata McGraw Hill Publishing Company	--
7	William Stallings	*Computer organization and Architecture*	Pearson Education	10th edition

Course Code	Name of the Course
216U01L401	**Competitive Programming Laboratory**

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	-	02	01	03
Credits Assigned	-	01	01	02

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	50	-	-	-	50

Course pre-requisites: Programming language, data structure and algorithm, Problem skill, mathematics

Course Objectives:

1. Algorithm Proficiency: Develop strong problem-solving skills and expertise in common algorithms and data structures.
2. Efficiency and Optimization: Teach efficient coding techniques and code optimization for quick problem solving.
3. Mathematical and Computational Concepts: Cover mathematical topics and concepts relevant to competitive programming.
4. Contest Preparation: Equip students with the skills and strategies needed to excel in coding competitions.

Course Outcomes (CO):

CO 1	Applying various problem-solving paradigms, enabling them to create and implement efficient algorithms for real-world challenges.
CO 2	Analyse and optimize algorithms using amortized analysis and bit manipulation, equipping them to tackle complex computational problems.
CO 3	Analyse and optimize algorithms using amortized analysis and bit manipulation, equipping them to tackle complex computational problems.
CO 4	Apply matrices, probability, game theory, and string algorithms in competitive programming.

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Problem Solving Paradigms**		08	CO 1
	1.1	**Complete search:** Generating subsets, Generating permutations, Backtracking, Pruning the search		
	1.2	**Greedy algorithms:** Coin problem, Scheduling, Minimizing sums, Data compression		
	1.3	**Dynamic programming:** Coin problem ,Longest increasing subsequence ,Paths in a grid Edit distance ,Counting tilings		
2	**Amortized analysis and Bit manipulation**		08	CO 2
	2.1	**Amortized analysis:** Two pointers method, Nearest smaller elements, Sliding window minimum.		
	2.2	**Bit manipulation:** Bit representation, Bit operations, Representing sets, Bit optimizations, Dynamic programming.		
#Self-learning: Range queries				
3	**Graphs and its applications**		04	CO 3
	3.1	**Basics of graphs:** Graph terminology, Graph representation, Applications.		
4	**Matrices, Game theory and String algorithms**		10	CO 4
	4.1	**Matrices:** Operations, Linear recurrences, Graphs and matrices **Probability:** Calculation, Events, Random variables, Markov chains, Randomized algorithms		
	4.2	**Game theory:** Game states, Nim game, Sprague–Grundy theorem		
#Self-Learning :Square root algorithms, Segment trees revisited, Geometry, Sweep line algorithms				
Total			**30**	--

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/ Year
1	Antti Laaksonen	*Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*	Springer	Second January 2020
2	Steven Halim and Felix Halim	*Competitive Programming 4 (Book1 and Book 2)*	Lulu Press, Inc	Second 2020
3	Introduction to Algorithms	*Thomas H . Cormen, Charles E. Leiserson*	MIT press	Fourth Edition 2022
4	Mahfudzah Othman, Naimah Mohd Hussin, Nora Yanti Che Jan	*Competitive Programming: Java and C++ (Questions and Solutions)*	Lulu Press, Inc	May 2019
5	https://cses.fi/problemset/, last retrieved on Apr 19, 2024	*CSES Problem Set*	-	-

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.

Course Code	Name of the Course
216U01L402	Analysis of Algorithms Laboratory

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	--	02	--	02
Credits Assigned	--	01	--	01

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	50	--	--	--	50

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 (CO):

CO 1	Analyse the asymptotic running time and space complexity of algorithms.
CO 2	Describe various algorithm design strategies to solve different problems and analyse complexity.
CO 3	Develop string matching techniques.
CO 4	Describe the classes P, NP, and NP-Complete.

Laboratory will consist of experiments/ tutorials covering entire syllabus of the course 216U01C402, ‘Analysis of Algorithms’. Students will be graded based on continuous assessment of laboratory work.

Course Code	Name of the Course
216U01L403	**Relational Database Management Systems Laboratory**

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	-	02	-	02
Credits Assigned	-	01	-	01

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	50	-	-	-	50

Course pre-requisites: Data Structure and programming knowledge.

Course Objectives:

The aim of this course is to equip the students with the knowledge and skills to design and program effective database systems. This includes understanding the Entity-Relationship (ER) approach to data modelling, the relational model of Database Management Systems (DBMS), and efficient database design through normalization. It introduces the relational algebra and query languages such as SQL to retrieve, manipulate, and manage data within a database. Further the concepts of Transaction Management, Concurrency Control and Recovery Techniques in databases striking a balance between providing a strong theoretical foundation for designing databases and practical skills for creating, querying, and implementing realistic databases.

Course Outcomes (CO):

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

CO 1	Comprehend the Characteristics of Relational Database Management Systems.
CO 2	Create Relational Database Designs Based on Entity-Relationship Models. Develop string matching techniques.
CO 3	Utilize SQL for Relational Database Operations.
CO 4	Analyse Advanced Database Concepts like indexing, hashing, query processing, query optimization, normalization.
CO 5	Apply Transaction Management, Concurrency Control, and Recovery Techniques

.

Laboratory will consist of experiments/ tutorials covering entire syllabus of the course 216U01C403, “Relational Database Systems”. Students will be graded based on continuous assessment of laboratory work.

Course Code	Name of the Course
216U01L404	Operating Systems Laboratory

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	-	02	-	02
Credits Assigned	-	01	-	01

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	50	-	-	-	50

Course pre-requisites: Basics of Computer Organization and architecture.

Course Objectives:

- To introduce basic concepts and functions of operating systems.
- To understand the concept of process, thread and resource management.
- To understand the concepts of process synchronization and deadlock
- To understand various Memory, I/O and File management techniques.
- To understand the designing and implementation of system software like Assembler.
- Macro pre-processor and linker loader

.

Course Outcomes (CO):

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

CO 1	Identify the different system programs and their utility and Explain the fundamental concepts of operating system with extension to Unix and Mobile OS.
CO 2	Illustrate and analyze the Process, threads, process scheduling and thread scheduling.
CO 3	Illustrate and analyze the Process, threads, process scheduling and thread scheduling
CO 4	Explain disk organization and file system structure with illustration of disk scheduling algorithms
CO 5	Understand Storage management with allocation, segmentation & virtual memory concepts

Laboratory will consist of experiments/ tutorials covering entire syllabus of the course 216U01C404, ‘Operating Systems’. Students will be graded based on continuous assessment of laboratory work.

Course Code	Name of the Course
216U01L405	Web Programming Laboratory

Teaching Scheme (Hrs./Week)	TH	P	TUT	Total
	-	02	02	04
Credits Assigned	-	01	02	03

Evaluation Scheme	Marks				
	LAB/TUT CA	CA (TH)		ESE	Total
		IA	ISE		
	75	-	-	-	75

Course pre-requisites: 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 (CO):

CO 1	Design dynamic web pages using various HTML tags.
CO2	Use CSS to prepare the layout of web pages.
CO 3	Apply JavaScript for validation in client side programming.
CO 4	Test the server-side pages that are integrated with PHP.
CO 5	Apply database operations by integrating SQL queries and session variables

Detailed Curriculum

Module No.	Unit No.	Contents	No of Hrs.	CO
1	**Basic of Web**		04	CO1
	1.1	Internet Fundamentals: Internet & DNS, HTTP		
	1.2	HTML5: 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. Table tags, Form Tags, Frame Tags, Executable content tags.		
2	**Basic of Frontend Styling**		04	CO2
	2.1	CSS3: 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	Advanced CSS3: CSS3 Media queries, Animation, Flexbox		
	2.3	Chrome Dev Tools: Console Tab, Sources Tab, Networks Tab, Elements Tab		
#Self Study: BootStrap				
3	**JavaScript**		07	CO3
	3.1	JavaScript Essentials: DOM, Modern , Introduction to JavaScript, Data Types, Operators, Control Flow, Arrays, and Functions,		
	3.2	Enhancing and Validating Forms		
	3.3	Advanced Practical JavaScript: Promises, DOM and Event, Objects, ECMAScript		
#Self Study: jQuery				
4	**PHP Programming**		07	CO4
	4.1	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		
	4.2	Version Control System : GIT Basics, Branches, Merging, Local Repository, Remote Repository, Pull Requests		
	4.3	Composer: require, install, update, package version control, packagist, autoload		
#Self Study : API Testing, REST, JSON API, Postman				
5	**Databases**		08	CO5
	5.1	PHP/MySQL Functions, Displaying queries in tables, Building Forms from queries, String and Regular		

Module No.	Unit No.	Contents	No of Hrs.	CO
		Expressions, Sessions, Cookies, Integration of complete web application and deployment.		
	5.2	MongoDB, PostgreSQL, PDO		
	5.3	Testing – PHP Unit		
#Self Study: Laravel, Symfony				
		Total	**30**	**--**

Reference Books*

Sr. No	Name/s of Author/s	Title of Book	Publisher	Edition/ Year
1	Steve Prettyman	Learn PHP 8 Using MySQL, JavaScript, CSS3, and HTML5	Apress	2nd / 2020
2	https://developer.mozilla.org/en-US/docs/Web/JavaScript, last retrieved on April 19, 2024			
3	https://developer.mozilla.org/en-US/docs/Glossary/HTML5, last retrieved on April 19, 2024			
4	https://www.w3schools.com/html/html_intro.asp, last retrieved on April 19, 2024			
5	https://developer.chrome.com/docs/devtools/, last retrieved on April 19, 2024			

*In addition to printed books, faculty can suggest (authentic) urls or e-books, e-contents etc.