University of Mumbai

Syllabus

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

From:
Academic Year 2019 – 20
(KJSCE 2018 CBGS Pattern)
Approved by Academic Council 18/05/2019
SY B. Tech. /COMP/ Revision 2.0

K. J. Somaiya College of Engineering, Vidyavihar, Mumbai – 77
(Autonomous College Affiliated to University of Mumbai)

It is notified for information of all concerned that the Board of Studies at its meeting held on April 30, 2019 and the subsequent meeting of the Academic Council held on May 18, 2019 amended the syllabus of SY B. Tech Computer Engineering and same be brought in to force from Academic Year 2019-20 with immediate effect.

Preamble

Academic Autonomy conferred by the University of Mumbai from the Academic Year 2014-15, gave us the freedom to develop and implement our own curriculum KJSCE2014 with features such as inclusion of choice based Interdisciplinary Course (IDC), Audit Courses, Add on Credit Courses, Exposure Courses, etc. Distinct assessment and evaluation methods were also designed based on focus of individual course. And 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, AICTE has taken necessary initiative in January 2018 by introducing model syllabus for undergraduate courses having a focus on the changing industrial scenario.

Our new revision in syllabus *KJSCE2018,* introduced from the academic year 2018-19, has been designed based on the revised AICTE guidelines as well as various accrediting bodies.

The said syllabus is a result of expert advice from members of Board of studies 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 *KJSCE2018* 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, Open Source Laboratory 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 two tests and internal assessment (IA) like quizzes, case studies, mini projects etc. These IA tools not only contribute to marks but also enables the student to learn 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

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)

PEO	Description
PEO 1	Solve problems in diverse fields using knowledge of Computer Engineering.
PEO 2	Excel in professional career, exhibit leadership qualities with ethics &soft skills
PEO 3	Pursue higher education, research or entrepreneurship, engage in professional development, adapt to emerging technologies

Program Outcomes (PO):

Engineering Graduates will be able to:

PO	Description
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, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
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 1	Apply acquired skills of Information Systems, Networking, Image processing in solving problems of varying complexities.
PSO 2	Pursue higher studies in the field of Computer Science & Engineering and be employable in industries

Acronym for category of courses

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	**T – 1**	Test – 1
AOAC	Add on Audit Course	**T – 2**	Test – 2
AVAC	Add on Value Audit Course	**CO**	Course Outcome
EX	Exposure Course	**PO**	Program Outcome
I	Interdisciplinary courses	**PSO**	Program specific Outcome

Acronyms used in Course code e.g. 2UCC301

Acronym Serially as per code	Definition
2	Second revision after autonomy KJSCE 2016 (First revision KJSCE 2014
U	Undergraduate
C	Department of Computer Engineering
C	Core Course
L	Laboratory Course
W	Workshop
T	Tutorial
X	Exposure Course
A	Audit Course
3	3- Semester 3 / 4- Semester 4
01	Course No.

Semester III
Credit and Examination Scheme

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
2UCC301	Integral Transform and Vector Calculus	3 – 0 – 1	4	3 – 0 – 1	4	BS
2UCC302	Data Structures$	3 – 0 – 0	3	3 – 0 – 0	3	PC
2UCC303	Computer Organization & Architecture	3 – 0 – 0	3	3 – 0 – 0	3	PC
2UCC304	Object Oriented Programming Methodology	1 – 0 – 2	3	1 – 0 – 2	3	PC
2UCC305	Discrete Mathematics	3 – 0 – 1	4	3 – 0 – 1	4	PC
2UCL301	Digital Design Laboratory	1 – 2 – 0	3	0 – 2 – 0	2	PC
2UCL302	Data Structures Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
2UCL303	Computer Organization & Architecture	0 – 2 – 0	2	0 – 1 – 0	1	PC
2UCL304	Object Oriented Programming Methodology Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
	Total	**14– 8 – 4**	**26**	**14 – 4 – 4**	**22**	
2USA3XX	Audit Course&	2 - 0- 0	2	--	--	AC

$- Common with IT Branch
& Completion of this course equivalent to 25 activity points

Semester III

Examination Scheme

Course Code	Course Name	Examination Scheme								
		Marks								
		CA								
		T – 1	T – 2	IA	ESE	TW	O%	P	P&O#	Total
2UCC301	Integral Transform and Vector Calculus	15	15	20	50	25	-	-		**125**
2UCC302	Data Structures$	15	15	20	50	-	-	-	-	**100**
2UCC303	Computer Organization & Architecture	15	15	20	50	-	-	-	-	**100**
2UCC304	Object Oriented Programming Methodology	15	15	20	50	-	-	-	-	**100**
2UCC305	Discrete Mathematics	15	15	20	50	25	-	-	-	**125**
2UCL301	Digital Design Laboratory	-	-	-	-	50	-	-	25	**75**
2UCL302	Data Structures Laboratory	-	-	-	-	25	-	-	25	**50**
2UCL303	Computer Organization & Architecture Laboratory	-	-	-	-	25	-	-	25	**50**
2UCL304	Object Oriented Programming Methodology Laboratory	-	-	-	-	25	-	-	25	**50**
	Total	**150**		**100**	**250**	**175**	**50**	**25**	**25**	**775**
2USA3XX	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
2UCC401	Probability, Statistics and Optimization Techniques$	3 – 0 – 1	4	3 – 0 – 1	4	BS
2UCC402	Analysis of Algorithms	3 – 0 – 0	3	3 – 0 – 0	3	PC
2UCC403	Relational Database Management Systems	3 – 0 – 0	3	3 – 0 – 0	3	PC
2UCC404	Theory of Automata with Compiler Design	3 – 0 – 1	4	3 – 0 – 1	4	PC
2UCL401	Python Programming Laboratory	0 - 4 - 0	4	0 – 2 – 0	2	PC
2UCL402	Analysis of Algorithms Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
2UCL403	Relational Database Management Systems Laboratory	0 – 2 – 0	2	0 – 1 – 0	1	PC
2UCL404	Mini Project	1 –2 – 0	3	1 – 2 – 0	3	PC
	Total	**13 – 10 – 2**	**25**	**13 – 6– 2**	**21**	
2USA4XX	Audit Course&	2 - 0 – 0	2	--	--	AC

$- Common with IT Branch
& Completion of this course equivalent to 25 activity points

Semester IV

Examination Scheme

Course Code	Course Name	Examination Scheme								
		Marks								
		CA			ESE	TW	O%	P	P&O#	Total
		T – 1	T – 2	IA						
2UCC401	Probability, Statistics and Optimization Techniques$	15	15	20	50	25	-	-	-	**125**
2UCC402	Analysis of Algorithms	15	15	20	50	-	-	-	-	**100**
2UCC403	Relational Database Management Systems	15	15	20	50	-	-	-	-	**100**
2UCC404	Theory of Automata with Compiler Design	15	15	20	50	25	-	-	-	**125**
2UCL401	Python Programming Laboratory	-	-	-	-	50			50	**100**
2UCL`402	Analysis of Algorithms Laboratory	-	-	-	-	25	-	-	25	**50**
2UCL403	Relational Database Management Systems Laboratory	-	-	-	-	25	-	-	25	**50**
2UCL404	Mini Project	-	-	-	-	50	-	-	50^	**100**
	Total	**120**		**80**	**200**	**200**	**-**	**-**	**150**	**750**
2USA4XX	Audit Course	2 - 0 – 0		-	-	-	-	-	-	-

% 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

Semester III

SY B. Tech Computer Engineering
(KJSCE 2018)

Course Code	Course Title							
2UCC301	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
	T-1	T-2	IA						
	15	15	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 irrotational 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		
	#Self Learning - Orthogonal and orthonormal set of functions over an interval, Surface Integral			
		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.	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 VidyarthiGruha, India	6th Edition 2012

Term-Work will consist of Tutorials covering 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							
2UCC302	Data structure[$]							
	TH			P		TUT		Total
Teaching Scheme(Hrs.)	03			02		00		03
Credits Assigned	03			01		00		04
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	15	15	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 and non-linear data structure in application development.
CO3: Describe concepts of advance data structures like set, map & dictionary.
CO4. 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**		**16**	**CO 2**
	2.1	**Linked List:** Introduction, 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	**Stack**: 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.		
		#Self Learning - application of stack evaluation of postfix and prefix expressions.		
	2.3	**Queues**: The Queue as an ADT, Queue operation, Array Representation of Queue, Linked 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.		
3	**Non-Linear data structures: Trees, Graph**		**12**	**CO 2**
	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 – Red-Black and Splay 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: Set, 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 : Exploring case studies on use of set, map and dictionary		
5	**Searching and Sorting**		**8**	**CO 4**
	5.1	**Sorting** :Sort Concept, Sort Stability , Bubble Sort, Shell Sort, Counting Sort		
		#Self learning : Buket 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							
2UCC303	Computer Organization and Architecture							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	03			--		--		03
Credits Assigned	03			--		--		03
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	15	15	20	50	--	--	--	--	100

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
At the end of successful completion of the course the student 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	Hrs.	CO
1	**Structure of a Computer System**		**0 4**	**CO1**
	1.1	Introduction of computer system and its sub modules, Basic organization of computer , Structure and Function, Brief history of computers, Von Neumann model		
	1.2	Introduction to buses, bus types and interconnection structures, PCI and SCSI bus		
2	**Arithmetic and Logic Unit**		**10**	**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	**Central Processing Unit**		**11**	**CO2**
	3.1	CPU architecture, Register organization, Instruction Sets: Operands and Operations , Instruction formats and addressing modes(Intel processor).,Basic instruction cycle, Instruction interpretation and sequencing		
	3.2	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		
	3.3	RISC v/s CISC processors, RISC and CISC Architecture, RISC pipelining, Case study on SPARC		
4	**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	**I/O Organization**		**05**	**CO4**
	5.1	External Devices , I/ O Modules		
	5.2	Programmed I/O, Interrupt driven I/O, DMA		
6	**Multiprocessor Configurations**		**04**	**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		
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	"Computer Organization and Architecture: Designing for performance",	Prentice Hall of India	8th Edition, 2003, ISBN 81 – 203 – 2962 – 7
2.	C. Hamacher, V. Zvonko, S. Zaky	"Computer Organization"	McGraw Hill	5th edition, 2002 ISBN 007-120411-3
3.	Kai Hwang & Bridggs	Computer Organization & Parallel Processing	McGraw Hill	International Editions 1985

Course Code	Course Title							
2UCC304	Object Oriented Programming Methodology							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	01			--		02*		03
Credits Assigned	01			--		02		03
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	15	15	20	50	--	--	--	--	100

* 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++. This course also provide difference 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 array, vector, 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**		**06**	**CO 1**
	1.1	Introduction, Procedural Programming Approach, Structured Programming Approach, Modular Programming Approach, OOP Approach		
	1.2	Objects and classes, Function overloading, Data abstraction and Encapsulation, Inheritance and Polymorphism, Function Overriding, Exceptions, Reuse, Coupling and Cohesion, Sufficiency Completeness and Primitiveness.		
2	**Class Object and method**		**12**	**CO 1**
	2.1	Class Object and Method: member, method, Modifier, Selector, constructer, destructor, iterator, State of an object, Method Overloading, Inheritance, Method Overriding, Final class, abstract class and method , Operator overloading		
	2.2	Comparing Object Oriented Concepts with Java and C++		
3	**Arrays and vectors**		**06**	**CO 2**
	3.1	Arrays: Arrays: 1D , 2D , Variable Length array, for-each with Array , Vectors: Vector Command line Arguments		
4	**Class Diagram**		**06**	**CO 3**
	4.1	Class Diagram		
	4.2	Implementing Aggregation and Association		
5	**Interfaces, Packages & Multithreading**		**15**	**CO4**
	5.1	Interfaces : variables in Interfaces, Extending an Interface, Difference between an Abstract class and an Interface, Packages: Creating Packages, Using Packages, Access Protection, Predefined packages , Wrapper class		
	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.		
	5.4	File Handling		
		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.	Sachin Malhotra,Saurabh Chaudhary	*Programming in JAVA*	Oxford University	2010
2.	Object Oriented Programming in C++	E Balagurusamy	Tata McGraw Hill	5th Edition, 2011
3.	Grady Booch,James Rumbaugh,Ivar Jacobson	Unified Modeling Language	Person Education	3rd Edition
4.	Let us C++	Yashwant Kanetkar	BPB publications	2nd Edition, 2003
5.	Ralph Bravaco,Shai simoson	*Java Programming from the Group up*	Tata McGraw-Hill	McGraw-Hill Edition
6.	Herbert schildt	*The complete Reference JAVA2*	Tata McGraw-Hill	2nd Edition 2002

Course Code	Course Title							
2UCC305	Discrete Mathematics							
	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
	T-1	T-2	IA						
	15	15	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**	**04**	**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**	**05**	**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**	**11**	**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, GF Fields, Primes and Related Congruence Equations- Primes, Primality Testing, Factorization, Quadratics Congruence, Chinese reminder theorem, Exponentiation and Logarithm.		
	#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
5				

Course Code	Course Title							
2UCL301	Digital Design Laboratory							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	1			2		-		3
Credits Assigned	-			2		0		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 technique and solve combinational circuits.
CO3. Design synchronous and asynchronous sequential circuits.
CO4. Implement digital networks using VHDL.

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 Algebric 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 compliment 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	

***Term work will consist of practical's covering entire syllabus of Digital Design.**
***Students will be graded based on continuous assessment of their term work**
***Practical and Oral examination based on laboratory experiments and entire syllabus**

Recommended Books:

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

Course Code	Course Title							
2UCL302	Data Structures Laboratory							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	--			02		--		02
Credits Assigned	--			01		--		01
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	--	--	--	--	25	--	--	25	50

- **Term-Work will consist of practical covering entire syllabus of "Data Structures" 2UCC302. 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" 2UCC303.**

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								
2UCEL303	Computer Organization and Architecture Laboratory								
	TH			P		TUT		Total	
Teaching Scheme (Hrs./Week)	--			02		--		02	
Credits Assigned	--			01		--		01	
Examination Scheme	Marks								
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	--	--	--	--	25	25	--	--	50

Term Work:
Note: The faculty should conduct 8-10 experiments based on the above syllabus
The programs should be implemented in C, C++ or Java

Term-Work will consist of Practical covering entire syllabus of "Computer Organization and Architecture" - 2UCEC303. Students will be graded based on Continuous Assessment of their term work.

Course Code	Course Title							
2UCL304	Object Oriented Programming Methodology Laboratory							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	--			02		--		02
Credits Assigned	--			01		--		01
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	--	--	--	--	25	--	--	25	50

Term Work will consist of Practical covering entire syllabus of "Object Oriented Programming Methodology" - 2UCC304. Students will be graded based on continuous assessment of their term work.
Practical which will demonstrate the object oriented concepts comparison in C++ and JAVA.

Semester IV

SY B. Tech Computer Engineering
(KJSCE-2018)

Course Code	Course Title							
2UCC401	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
	T-1	T-2	IA						
	15	15	20	50	25	--	--	--	125

$- Common with IT Branch
* 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 Non Linear 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	**Correlation and Regression**		06	CO 1
	1.1	Correlation, Co-variance, Karl Pearson Coefficient of Correlation & Spearman's Rank Correlation Coefficient		
	1.2	Regression Coefficients , lines of regression & logistic regression		
2	**Probability and Probability Distribution**		12	CO 2
	2.1	Conditional Probability, Bayes' theorem, Joint Probability		
	2.2	Discrete and Continuous Probability Distribution		
	2.3	Binomial Distribution, Poisson Distribution		
	2.4	Uniform Distribution, Normal Distribution, Exponential Distribution		
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		
	#Self Learning Topic- Fitting First degree, second degree and exponential curve to the given bivariate data.			
		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.	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

Term-Work will consist of Tutorials covering 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 softwarein the Laboratory.

Course Code	Course Title							
2UCC402	Analysis of Algorithms							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	03			--		--		03
Credits Assigned	03			--		--		03
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	15	15	20	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 problem. 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-Kruskals 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	
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							
2UCC403	Relational Database Management Systems							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	03			--		--		03
Credits Assigned	03			--		--		03
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	15	15	20	50	--	--	--	--	100

Course prerequisites:

NIL.

Course Objectives:

The objective of the course is to understand the need of database systems, using ER model to design the data. Convert the ER model data to relational data model. Using SQL to Design, implement and manage the relational model in DBMS. The course also gives introduces to different concurrency and recovery techniques of data.

Course Outcomes

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

CO1: Relate database management system concepts and design Entity-Relationship model for different database application

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

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

CO4. Apply normalization to improve design of the database

CO5. Demonstrate 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**		06	CO1
	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		
3.	**Data model: Relational Data Model and Relational Algebra**		08	CO2
	3.1	Introduction , Data Manipulation , Data Integrity, Advantages of the Relational Model		
	3.2	Mapping EER Model to Relational Model		
	3.3	Relational Algebra , Relational Algebra Queries		
4	**Structured Query Language (SQL)**		09	CO3
	4.1	Overview of SQL , Data Definition Commands		
	4.2	Set operations, aggregate function, null values, Data Manipulation commands		
	4.3	Data Control commands , Views in SQL		
	4.4	Nested and complex queries,		
	4.5	Domain Constraints, Referential integrity,		
	4.6	Assertions Trigger.		
	4.7	Security and authorization in SQL		
	4.8	Introduction to query processing		
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**		10	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, Multiversion 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							
UCC404	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
	T-1	T-2	IA						
	15	15	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, Push Down Automata, Un-decidability and Intractable Problems. It aims to understand the design of computing machine 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, simplified notations (state transition diagram, transition table)		
	1.3	Language of a DFA. NFA -Formal definition, Language of an NFA.		
	1.4	FA with epsilon-transitions, Eliminating epsilon-transitions		
	1.5	Equivalence of DFAs and NFAs, An Application.		
2	**Regular Languages**		09	CO1
	2.1	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, M hill-Nerode Theorem		
	2.4	DFA Minimization.some decision properties of Regular languages emptiness, finiteness, membership, equivalence of two DF As or REs, Finite automata with output		
	2.5	Regular sets, Regular Expression, Some closure properties of Regular languages -Closure under Boolean operations, reversal, homomorphism, inverse homomorphism, etc..		
	2.6	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, . Chomsky hierarchy		
	3.4	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		

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, Multistack DPDAs & NPDAs and CFLs. Languages of DPDAs, NPDAs, and ambiguous grammars.		
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 -Multitape TMs, Nondeterministic TMs. TMs with semi-infinite tapes, Multistack 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 Un-decidability, 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	
2.	J.C.Martin,	"Introduction to languages and the Theory of Computation"	TMH	
3.	Michael Sipser	"Theory of Computation"	Cengage Learning	
4.	O.G.Kakde	"Theory of Computation"	LP.	

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

Course Code	Course Title							
2UCL401	Python Programming Laboratory							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	--			04		--		04
Credits Assigned	--			02		--		02
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	--	--	--	--	25	--	--	50	75

Course prerequisites:

Students should be familiar with concepts of programming language.

Course Objectives

- Basics of Python programming
- Object Oriented Programming using Python
- Files Handling in Python
- GUI Programming and Databases operations in Python

Course Outcomes

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

CO1: Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python

CO2: Interpret different Decision Making statements, Functions, Object oriented programming in Python

CO3: Understand and summarize different File handling operations

CO4: Explain how to design GUI Applications in Python and evaluate different database operations

Module No.	Unit No.	Details	Hrs.	CO
1	**Introduction to Python**			
		What Is Python, History of Python, Features of Python, Installing Python, Running Python, Comparing Python JPython and Some Nomenclature	**02**	**CO1**
2	**Core Python Programming**			
	2.1	**Data Structures and Control Statements** Comments, Operators, Variables and Assignment, Numbers, Strings, Lists and Tuples, Dictionaries, Code Blocks Use Indentation if Statement, while Loop, for Loop and the range() Built-in Function, User Defined Functions, Errors and Exceptions, Classes, Modules, Packages	**09**	**CO1, CO2**
	2.2	**Files and Input/ Output** File Objects, File Built-in Function, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules	**04**	**CO3**
	2.3	**Errors And Exceptions** What Are Exceptions?, Exceptions in Python, Detecting and Handling Exceptions, Exceptions as Strings, Exceptions as Classes, Raising Exceptions Assertions, Standard Exceptions, Creating Exceptions, Exceptions and the sys Module	**03**	**CO2**
3	**GUI Programming with Tkinter**			
		Introduction, Tkinter and Python Programming, Tkinter Examples	**07**	**C04**
4	**Regular Expressions**			
		Introduction/Motivation,Special Symbols and Characters for Res**,** REs and Python**,** Regular Expression Adventures	**02**	**CO4**
6	**Database Access**			
		Python's Database Connectivity, Types of Databases Used with Python, Mysql database Connectivity with Python, Performing Insert, Deleting & Update operations on database	**03**	**C04**
TOTAL			**30**	

The students are expected to practice the Theory taught during practical hours.

Recommended Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
12.	Wesley J Chun	Core Python Applications Programming	Pearson Publication.	3rd Edition
13.	David Beazley, Brian K. Jones	Python Cookbook: Recipes for Mastering Python 3	O'REILLY	3rd Edition
14.	Lutz M	Learning Python: Powerful Object-Oriented Programming	O'REILLY	5th Edition
15.	John M. Zelle, Guido van Rossum	Python Programming: An Introduction to Computer Science	Franklin, Beedle & Associates	3rd Edition

Term-Work will consist of Practical covering entire syllabus of "Python Programming Laboratory" - 2UCL401 Students will be graded based on continuous assessment of their term work

Course Code	Course Title							
2UCL402	Analysis of Algorithms Laboratory							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	--			02		--		02
Credits Assigned	--			01		--		01
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	--	--	--	--	25	--	--	25	50

Term-Work will consist of Practical covering entire syllabus of "Analysis of Algorithms" - 2UCC402. Students will be graded based on continuous assessment of their term work

Note: The programs should be implemented in C, C++ or Java

Course Code	Course Title							
2UCL403	Relational Database Management Systems Laboratory							
	TH			P		TUT		Total
Teaching Scheme (Hrs./Week)	--			02		--		02
Credits Assigned	--			01		--		01
Examination Scheme	Marks							
	CA			ESE	TW	O	P	P&O	Total
	T-1	T-2	IA						
	--	--	--	--	25	--	--	25	50

Term-Work will consist of Practical covering entire syllabus of "Relational Database Management Systems" - 2UCC403 Students will be graded based on continuous assessment of their term work

Course Code	Course Title						
2UCL404	Mini Project						
	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
	T-1	T-2	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. Define the problem statement and scope of problem

CO2. Identify various hardware and software requirements for problem solution

CO3. Implement and test the hardware/ software algorithms to meet the desired specifications.

CO4. Analyze, interpret results and correspondingly modify the designed system to get the desired results.

CO5. Prepare a technical report based on the Mini project.

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

Term Work and Oral:

The mini project is a group project. Interdisciplinary projects are also permitted. Each project will be assigned one faculty member as a supervisor. There will be continuous assessment and progress report of the project 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.