AC 29/4/2013 Item no. 4.74

UNIVERSITY OF MUMBAI

Revised Syllabus

Program- Bachelor of Engineering

Course -Computer Engineering

(Second Year – Sem. III & IV)

Under

FACULTY OF TECHNOLOGY

(As per Credit Based Semester and Grading System from 2013-14)

From Dean's Desk:

To meet the challenge of ensuring excellence in engineering education, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education.The major emphasis of accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Technology of University of Mumbai has taken a lead in incorporating philosophy of outcome based education in the process of curriculum development.

Faculty of Technology, University of Mumbai, in one of its meeting unanimously resolved that, each Board of Studies shall prepare some Program Educational Objectives (PEO's) and give freedom to affiliated Institutes to add few (PEO's) and course objectives and course outcomes to be clearly defined for each course, so that all faculty members in affiliated institutes understand the depth and approach of course to be taught, which will enhance learner's learning process. It was also resolved that, maximum senior faculty from colleges and experts from industry to be involved while revising the curriculum. I am happy to state that, each Board of studies has adhered to the resolutions passed by Faculty of Technology, and developed curriculum accordingly. In addition to outcome based education, semester based credit and grading system is also introduced to ensure quality of engineering education.

Semester based Credit and Grading system enables a much-required shift in focus from teacher-centric to learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on continuous evaluation which will enhance the quality of education. University of Mumbai has taken a lead in implementing the system through its affiliated Institutes and Faculty of Technology has devised a transparent credit assignment policy and adopted ten points scale to grade learner's performance.Credit assignment for courses is based on 15 weeks teaching learning process, however content of courses is to be taught in 12-13 weeks and remaining 3-2 weeks to be utilized for revision, guest lectures, coverage of content beyond syllabus etc.

Credit and grading based system was implemented for First Year of Engineering from the academic year 2012-2013. Subsequently this system will be carried forward for Second Year Engineering in the academic year 2013-2014, for Third Year and Final Year Engineering in the academic years 2014-2015 and 2015-2016 respectively.

Dr. S. K. Ukarande
Dean,
Faculty of Technology,
Member – Management Council, Senate, Academic Council
University of Mumbai, Mumbai

Preamble:

The engineering education in India in general is expanding in manifolds. Now, the challenge is to ensure its quality to the stakeholders along with the expansion. To meet this challenge, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education and reflects the fact that in achieving recognition, the institution or program of study is committed and open to external review to meet certain minimum specified standards. The major emphasis of this accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Technology of University of Mumbai has taken a lead in incorporating philosophy of outcome based education in the process of curriculum development.

The Program Educational Objectives finalized for undergraduate program in Computer Engineering are listed below:

1. To prepare Learner's with a sound foundation in the mathematical, scientific and engineering fundamentals
2. To prepare Learner's to use effectively modern tools to solve real life problems
3. To equip Learner's with broad education necessary to understand the impact of computer Technology in a global and social context
4. To encourage , motivate and prepare Learner's for Lifelong-learning
5. To inculcate professional and ethical attitude, good leadership qualities and commitment to social responsibilities

In addition to above 2 to3 more program educational objectives of their own may be added by affiliated Institutes. The Program outcomes are the skills and ability that Learner will demonstrate upon completion of undergraduate degree program in Computer Engineering. Few may be listed as follows:

1. Ability to effectively apply knowledge of computing and mathematics to computer science problems.
2. Ability to design, implement and evaluate computer-based components, systems, processes or programs to meet desired needs and specifications.
3. Ability and skills to effectively use state-of-the-art techniques and computing tools for analysis, design, and implementation of computing systems.
4. Ability to function effectively as a member of a team assembled to undertake a common goal.
5. An understanding of professional, ethical, legal, security, and social issues and responsibilities.
6. Ability to communicate effectively to both technical and non-technical audiences.
7. The ability to successfully pursue professional development thru lifelong learning

In addition to Program Educational Objectives, for each course of undergraduate program, Course Objectives and expected outcomes from learner's point of view are also included in the curriculum to support the philosophy of outcome based education. In order to achieve outcome 1,2,and 3 a major emphasis is planned towards designing Laboratory courses third year onwards. I believe strongly that small step taken in right direction will definitely help in providing quality education to the stake holders.

Dr. Prachi Gharpure

Chairperson, Adhoc Board of Studies in Computer Engineering

University of Mumbai, Mumbai

Program Structure for B.E. Computer Engineering

Second Year (Computer) (Semester III)

(REV 2012)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/ Pract	Tut	Total
CSC301	Applied Mathematics III*	4	-	1#	4	-	1	5
CSC 302	Object Oriented Programming Methodology*	4	2	-	4	1	-	5
CSC303	Data Structures	4	2	-	4	1	-	5
CSC304	Digital Logic Design and Analysis	3	2	-	3	1	-	4
CSC305	Discrete Structures	4	-	-	4	-	-	4
CSC306	Electronic Circuits and Communication Fundamentals	4	2	-	4	1	-	5
	Total	**23**	**8**	**1**	**23**	**4**	**1**	**28**

Course Code	Course Name	Examination Scheme							
		Internal Assesment							
		Internal Assesment			End Sem Exam	Exam Duration (in Hrs)	TW	Pract / oral	Tot
		Test 1	Test 2	Avg					
CSC301	Applied Mathematics III*	20	20	20	80	03	25!	-	125
CSC302	Object Oriented Programming Methodolgy*	20	20	20	80	03	25	25	150
CSC303	Data Structures	20	20	20	80	03	25	25	150
CSC304	Digital Logic Design and Analysis	20	20	20	80	03	25	-	125
CSC305	Discrete Structures	20	20	20	80	03	-	-	100
CSC306	Electronic Circuits and Communication Fundamentals	20	20	20	80	03	25	25	150
	Total	**-**	**-**	**120**	**480**	**-**	**125**	**75**	**800**

* Common Subjects with IT # Tutorial to be taken class wise ! **Tutorials will be evaluated as Term work**

Program Structure for B.E. Computer Engineering

Second Year (Computer) (Semester IV)

(REV 2012)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/ Pract	Tut	Total
CSC401	Applied Mathematics IV*	4	-	1	4	-	1#	5
CSC402	Analysis of Algorithms	4	2	-	4	1	-	5
CSC403	Computer Organization and Architecture*	4	2	-	4	1	-	5
CSC404	Data Base Management systems	4	2	-	4	1	-	5
CSC405	Theoretical Computer Science	4	-		4	-	-	4
CSC406	Computer Graphics	3	2	-	3	1	-	4
	Total	**23**	**8**	**1**	**23**	**4**	**1**	**28**

Course Code	Course Name	Examination Scheme							
		Internal Assesment							
		Internal Assesment			End Sem Exam	Exam Duration (in Hrs)	TW	Prac / oral	Tot
		Test 1	Test 2	Avg					
CSC401	Applied Mathematics IV*	20	20	20	80	03	25!	-	125
CSC402	Analysis of Algorithms	20	20	20	80	03	25	25	150
CSC403	Computer Organization and Architecture*	20	20	20	80	03	25	25	150
CSC404	Data Base Management systems	20	20	20	80	03	25	25	150
CSC405	Theoretical Computer Science	20	20	20	80	03	-	-	100
CSC406	Computer Graphics	20	20	20	80	03	25	25	150
	Total	**-**	**-**	**120**	**480**	**-**	**125**	**100**	**825**

* Common Subjects with IT # Tutorial to be taken class wise

! **Tutorials will be evaluated as Term work**

Course Code	Course Name	Credits
CSC301	**Applied Mathaematics III**	05

Objectives:

1) Complex Variable (2) Laplace Transform (3) Fourier Series (4) Discrete Structures (5) Z-transform

These topics involve the study of analytic function and mapping of complex function, Laplace transform, Inverse Laplace transform and application of Laplace transform to solve differential equations, finding Fourier series, Sine and cosine Fourier integral and Z-transform. These topics help them to solve many engineering problems arising in course of their further studies and also while working in the practical life situations.

Outcomes:

Students in this course will apply the Procedure and methods to solve technical problems.

Details of the Syllabus:-

Module	Topics	Hrs
01	**Complex Variable & mapping** 1.1 Functions of a complex variable, Analytic functions, Cauchy-Riemann equations in Cartesian co-ordinates, Polar co-ordinates. 1.2 Harmonic functions, Analytic method and Milne Thomson methods to find f(z), Orthogonal trajectories. 1.3 Conformal Mapping, Linear, Bilinear transformations, Cross ratio, fixed points and standard transformation such as rotation and magnification, invertion, translation.	**(10)**
02	**Laplace Transform** 2.1 Introduction, Definition of Laplace transform, Laplace transform of constant, trigonometrical, exponential functions. 2.2 Important properties of Laplace transform: First shifting theorem, Laplace transform of $L\{t^n f(t)\}$, $L\{f(t)/t\}$, $L\left\{\frac{d^n f(t)}{dt^n}\right\}$, $L\{\int_0^t f(u)du\}$, $L\{f(at)\}$ without proof. 2.2Unit step function, Heavi side function, Dirac-delta function, Periodic function and their Laplace transforms, Second shifting theorem. 2.3Inverse Laplace transform with Partial fraction and Convolution theorem (without proof). 2.4 Application to solve initial and boundary value problem involving ordinary differential equations with one dependent variable and constant coefficients.	**(10)**

03	**Fourier series** 3.1 Dirichlet's conditions, Fourier series of periodic functions with period 2π and 2L. 3.2 Fourier series for even and odd functions. 3.3 Half range sine and cosine Fourier series, Parsevel's identities (without proof). 3.4Orthogonal and Ortho-normal functions, Complex form of Fourier series. 3.5 Fourier Integral Representation.	**(10)**
04	**Vector Algebra and Calculus** 4.1 Vector Algebra: Scalar and vector product of three and four Vectors and their properties. 4.2 Vector Calculus: Vector differential operator ∇, Gradient of a scalar point function, Diversions and Curl of Vector point function, $\nabla (u\, v)$, $\nabla.(\phi\vec{u}), \nabla\times(\phi\vec{u}), \nabla\times(\vec{u}\times\vec{v})$. 4.3 Vector Integration: Line integral; conservative vector field,Green's theorem in a plane (Without proof) 4.4 GaussDivergence theorem & Stokes' theorem (Without proof and no problems on verification of above theorems).	**(10)**
05	**Z transform** 5.1 Z-transform of standard functions such as $Z(a^n)$, $Z(n^p)$. 5.2 Properties of Z-transform :Linearity, Change of scale, Shifting property, Multiplication of K, Initial and final value, Convolution theorem (all without proof) 5.3 Inverse Z transform: Binomial Expansion and Method of Partial fraction.	**(8)**

Term work:

Term work shall consist of minimum four SCILAB practicals and six tutorials.

SCILAB practicals	:	10 marks
Tutorials	:	10 marks
Attendance	:	05 marks
Total	**:**	**25 marks**

Text Books:

- Higher Engineering Mathematics by Grewal B. S. 38th edition, Khanna Publication 2005.
- Advanced Engineering Mathematics by Kreyszig E. 9th edition, John Wiley.
- A Text Book of Applied Mathematics Vol. I & II by P.N.Wartilar & J.N.Wartikar, Pune, Vidyarthi Griha Prakashan., Pune.
- Discrete and Combinational Mathematics by Ralph P. Crimaldi, B Y Ramana.

References:

- Advanced Engg. Mathematics by C. Ray Wylie & Louis Barrett.TMH International Edition.
- Mathematical Methods of Science and Engineering by Kanti B. Datta, Cengage Learning.
- Lapplace Treansforms by Murry R. Spieget, Schaun's out line series-McGraw Hill Publication.
- Discrete mathematics by ERIL FOSSETT, Wiley India.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC302	Object Oriented Programming Methodology (OOPM)*	05

Course Objectives

1. To understand Object oriented concepts like data abstraction, encapsulation, etc.
2. To solve the real world scenarios using top down approach.
3. To understand various Java programming constructs.

Course Outcomes

1. Students will be able to solve computational problems using basic constructs like if-else, control structures, array, strings.
2. Student can understand how to model real world scenario using class diagram.
3. Students will exhibit communication between 2 objects using sequence diagram.
4. Students will be able to implement relationships between classes.
5. Students will be able to demonstrate various collection classes.
6. The students will be able to demonstrate programs on exceptions, multithreading and applets.

Sr. No	Topic	No of Hours
1	Programming Approach from proceduaral to Object Orientation OO methodologies: Grady Booch Methodology of OO development	4
2	OO Concepts: Object, Class, Encapsulation or information hiding, Inheritance, Polymorphism, Message communication, Abstraction, Reuse, Coupling and Cohesion, Sufficiency Completeness and Primitiveness, Meta class	5
3	Object Oriented Programming: Java Evolution: History, How java differs from others Overview of Java language: Introduction, Installing and implementing Java, JVM	3
4	Constants, variables and data types Operators and Expressions Revision of Branching and looping	6
5	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	6

6	Classes and Relationships : Implementation of Association and Aggegation using simple scenarios	2
7	Array, String, Vector	6
8	Interfaces : variables in Interfaces, Extending an Interface, Difference between an Abstarct class and an Interface	4
9	Multithread programming	4
10	Grouping of classes for deployment and reuse: Built-in Packages: java.lang: wrapper classes java.util: ArrayList and LinkedList Creating and using User defined packages	3
11	Managing Error and Exception	3
12	Applet programming	2

Suggested list of Programming Assignments /Laboratory Work

Divide laboratory work into 3 parts

A. Basic Java structural components and Conditional and control statements:

- To demonstrate the use of command line argument.
- To demonstrate various ways of accepting data through keyboard.
- To understand the working of an array.
- To understand string class and demonstrate its various functions.

B. Perform following practical on some case study like Banking Application, Library Application etc.

- Find out classes, objects and their properties.
- Create and display objects found in above.
- Add methods to classes and implement.
- Refine above objects by adding constructors and local variables.
- Show communication between the objects by calling instance of one object from another class.
- Find relationships like inheritance, association, aggregation, composition.
- Implement above relationships.

C.

- To implement user defined exceptions in Java.
- Demonstrate the use collection classes like ArrayList/LinkedList/HashSet/TreeSet/Map.

- To illustrate Multithreading in Java.
- Simple programs on Applets and AWT.

TermWork:

Students will submit Term Work in the form of a journal that will include at least 15 programming assignments. Each programming assignment will consist of an algorithm or class diagram/sequence diagram (if applicable), program listing with proper documentation and snapshot of the output.

Practical Examination will be based on the term work and questions will be asked to judge understanding of the assignments at the time of the examination.

The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.

Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 5 Marks (Assignment) + 5 (Attendance (theory+practical))

Practical Exam will based on above syllabus

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Text Books:

1. Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” ,Tata McGraw-Hill
2. Grady Booch, Object Oriented Analysis and Design ;
3. Jaime Nino, Frederick A. Hosch, ‘An introduction to Programming and Object Oriented Design using Java’, Wiley Student Edition.

Reference Books:

1. Java: How to Program, 8/e, Dietal, Dietal, PHI
2. Grady Booch, James Rumbaugh, Ivar Jacobson, “The Unified Modeling Language User Guide”, Pearson Education
3. Sachin Malhotra, Saurabh Chaudhary “Programming in Java”, Oxford University Press, 2010

Subject Code	Subject Name	Credits
CSC303	**Data Structures (DS)**	5

Course Objectives

1. To teach efficient storage mechanisms of data for an easy access.
2. To design and implementation of various basic and advanced data structures.
3. To introduce various techniques for representation of the data in the real world.
4. To develop application using data structures.
5. To teach the concept of protection and management of data.
6. To improve the logical ability

Course Outcomes

1. Student will be able to choose appropriate data structure as applied to specified problem definition.
2. Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.
4. Students will be able to use linear and non-linear data structures like stacks , queues , linked list etc.

Module	Detailed content	Hours
01	**Introduction to Data Structure** Types of Data Structure, Arrays, Strings, Recursion, ADT (Abstract Data type),Concept of Files,Operations with files, types of files	05
Linear Data Structure		
02	**Linked List** Linked List as an ADT, Linked List Vs. Arrays, Memory Allocation & De-allocation for a Linked List, Linked List operations, Types of Linked List, Implementation of Linked List, Application of Linked List- polynomial, sparse matrix.	10
03	**STACK** The Stack as an ADT, Stack operation, Array Representation of Stack, Link Representation of Stack, Application of stack – Recursion, Polish Notation	04
04	**Queues** The Queue as an ADT, Queue operation, Array Representation of Queue, Linked Representation of Queue, Circular Queue, Priority Queue, & De-queue, Application of Queues – Johnsons Algorithm, Simulation	05

Non-linear Data Structure		
05	**Trees** Basic trees concept, Binary tree representation,Binary tree operation, Binary tree traversal, Binary search tree implementation, Thread Binary tree, The Huffman Algorithm, Expression tree, Introduction to Multiway search tree and its creation(AVL, B-tree, B+ tree)	10
06	**Graphs** Basic concepts, Graph Representation, Graph traversal (DFS & BFS)	04
Sorting AND Searching		
07	**Sorting :** Sort Concept, Shell Sort, Radix sort, Insertion Sort, Quick Sort, Merge Sort,Heap Sort, **Searching :** List Search,Linear Index Search, Index Sequential Search Hashed List Search, Hashing Methods , Collision Resolution	10

Text Books:

1. Data Structures A Psedocode Approach with C, Richard F. Gilberg & Behrouz A. Forouzan, second edition, CENGAGE Learning.
2. Data Structures using C, Reema Thareja, Oxford University press.
3. Introduction to Data Structure and its Applications Jean-Paul Tremblay, P. G. Sorenson

Reference Books:

1. Data Structures Using C & C++, Rajesh K. Shukla, Wiley- India.
2. Data Structures Using C, ISRD Group, Second Edition, Tata McGraw-Hill
3. Data Structure Using C, Balagurusamy
4. C & Data Structures, Prof. P.S. Deshpande, Prof. O.G. Kakde, Dreamtech press.
5. Data Structures, Adapted by: GAV PAI, Schaum's Outlines

Termwork:

Term work should consist of at least 12 experiments.
Journal must include at least 2 assignments.
The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 5 Marks (Assignment) + 5 (Attendance (theory+practical))
Practical exam will be based on the above syllabus.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Suggested Experiments:

Note: Students are required to complete 12 experiments. The star (*) marks experiments are mandatory.

Linked List
• Implementations of Linked Lists menu driven program. • * Implementation of different operations on linked list – copy, concatenate, split, reverse, count no. of nodes etc • Representation of Sparse matrix using multilinked structure. Implementation of sparse matrix multiplication. • Implementation of polynomials operations (addition, subtraction) using Linked List. • Implementations of Linked Lists menu driven program (stack and queue) • Implementations of Double ended queue using Linked Lists. • Implementation of Priority queue program using Linked Lis
Stack
• Implementations of stack menu driven program • Implementation of multistack in one array. • * Implementations of Infix to Postfix Transformation and its evaluation program. • Implementations of Infix to Prefix Transformation and its evaluation program. • Simulation of recursion
Queue
• Implementations of circular queue menu driven program • * Implementations of double ended queue menu driven program • Implementations of queue menu driven program • Implementation of Priority queue program using array. • Implementation of Johnsons Algorithm • Implementation of Simulation Problem
Tree

- *Implementations of Binary Tree menu driven program
- Implementation of Binary Tree Traversal program.
- *Implementation of construction of expression tree using postfix expression.
- Implementations of Huffman code construction
- Implementations of BST program
- Implementation of various operations on tree like – copying tree, mirroring a tree, counting the number of nodes in the tree, counting only leaf nodes in the tree.
- Implementations of B-tree menu driven program
- Implementations of B+ tree program
- Implementation of Preorder traversal of a threaded binary tree.
- Implementations of AVL Tree menu driven program

Sorting

- *Implementations of Shell sort, Radix sort and Insertion sort menu driven program
- Implementations of Quick Sort, Merge sort and Heap Sort menu driven program

Searching

- *Implementations of searching methods (Index Sequential, Interpolation Search) menu driven program
- Implementation of hashing functions with different collision resolution techniques

Graph

- * Implementations of Graph menu driven program (DFS & BSF)

Subject Code	Subject Name	Credits
CSC304	Digital Logic Design and Analysis	4

Course Objective:

1. To provide concepts that underpins the disciplines of digital electronics and microprocessor systems.
2. To provide the concept of modeling Combinational and sequential circuits.
3. To provide basic knowledge of how digital building blocks are described in VHDL.

Course Outcomes:

1. Binary and hexadecimal calculations and conversions.
2. Designing of combinational circuits.
3. Design synchronous and asynchronous sequential circuits.
4. Translate real world problems into digital logic formulations.
5. Construct test and debug digital networks using VHDL.
6. Learners will show awareness about TTL and CMOC Logic

Module	Detailed Contents	Hours
1	Number Systems and Codes: Revision of Binary, Octal, Decimal and Hexadecimal number Systems and their conversion, Binary Addition and Subtraction (1's and 2's complement method), Gray Code, BCD Code, Excess-3 code, ASCII Code, Error Detection and Correction Codes.	05
2	Boolean Algebra and Logic Gates: Theorems and Properties of Boolean Algebra, Standard SOP and POS form, Reduction of Boolean functions using Algebric method, K -map method (2,3,4 Variable), and Quine-McClusky Method. NAND-NOR Realization. Basic Digital Circuits: NOT,AND,OR,NAND,NOR,EX-OR,EX-NOR Gates, Logic Families: Terminologies like Propagation Delay, Power Consumption, Fan in and Fan out etc. with respect to TTL and CMOS Logic and comparison.	10
3	Combinational Logic Design: Introduction, Half and Full Adder, Half and Full Subtractor, Four Bit Binary Adder, one digit BCD Adder, Four Bit Binary Subtractor (1's and 2's compliment method), code conversion, Multiplexers and Demultiplexers, Decoders, One bit, Two bit ,4-bit Magnitude Comparator .	08
4	Sequential Logic Design: Concept of Multivibrators: Astable, Monostable and Bistable multivibrators, Flip Flops:SR, D, JK, JK	10

	Master Slave and T Flip Flop, Truth Tables and Excitation Tables, Flip-flop conversion. sequential circuit analysis , construction of state diagrams. Counters: Design of Asynchronous and Synchronous Counters, Modulo Counters, UP- DOWN counter . Shift Registers: SISO, SIPO,PIPO,PISO, Bidirectional Shift Register, Universal Shift Register, Ring and Johnson Counter. Pseudorandom sequence generator.	
5	Functional Simulation , Timing Simulation, Logic synthesis, Introduction to VHDL, Framework of VHDL program(Syntax and programming to be done only during Practicals), Introduction to CPLD and FPGA	03

Text Books:

1. R. P. Jain, "Modern Digital Electronics", Tata McGraw Hill.
2. Yarbrough John M. , "Digital Logic Applications and Design ", Cengage Learning
3. J. Bhasker." VHDL Primer", Pearson Education

Reference Books:

1. M. Morris Mano, "Digital Logic and computer Design", PHI.
2. Douglas L. Perry, "VHDL Programming by Example", Tata McGraw Hill.
3. Donald p Leach, Albert Paul Malvino,"Digital principles and Applications",Tata McGraw Hill.

Termwork:

Term work should consist of at least 12 experiments out of which at least 2 to be VHDL based.
Journal must include at least 2 assignments.
The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 05 Marks (Assignment) + 05 (Attendance (theory+practical))

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC305	Discrete Structures	4

Course Objectives

1. To assimilate discrete mathematical concepts.
2. Introducing discrete maths as basic foundation of analysis and applications like communication,

Course Outcomes

1. Ability to reason logically.
2. Ability to understand use of functions, graphs and trees in programming applications.
3. Understand use of groups and codes in Encoding-Decoding.
4. Express recursive functions of other subjects like Data Structures as recurrence relation.

Module	Detailed content	Hours
01	**Set Theory** • Sets, Venn diagrams, Operations on Sets • Laws of set theory, Power set and Products • Partitions of sets, The Principle of Inclusion and Exclusion	05
02	**Logic** • Propositions and logical operations, Truth tables • Equivalence, Implications • Laws of logic, Normal Forms • Predicates and Quantifiers • Mathematical Induction	06
03	**Relations, Digraphs and Lattices** • Relations, Paths and Digraphs • Properties and types of binary relations • Manipulation of relations, Closures, Warshall's algorithm • Equivalence and partial ordered relations • Posets and Hasse diagram • Lattice	08

04	**Functions and Pigeon Hole Principle** • Definition and types of functions: Injective, Surjective and Bijective • Composition, Identity and Inverse • Pigeon-hole principle	**06**
05	**Generating Functions and Recurrence Relations** • Series and Sequences • Generating functions • Recurrence relations • Recursive Functions: Applications of recurrence relations e,g, Factorial, Fibonacci, Binary search, Quick Sort etc.	**06**
06	**Graphs and Subgraphs** • Definitions, Paths and circuits: Eulerian and Hamiltonian • Planer graphs, Graph coloring • Isomorphism of graphs • Subgraphs and Subgraph isomorphism	**06**
07	**Trees** • Trees and weighted trees • Spanning trees and minimum spanning tree • Isomorphism of trees and sub trees • Prefix codes	**05**
08	**Algebraic Structures** • Algebraic structures with one binary operation: semigroup, monoids and groups • Product and quotient of algebraic structures • Isomorphism, Homomorphism and Automorphism • Cyclic groups, Normal subgroups • Codes and group codes	**06**

Text Books:

1. Kenneth H. Rosen. ”Discrete Mathematics and its Applications”, Tata McGraw-Hill.
2. Bernad Kolman, Robert Busby, Sharon Cutler Ross, Nadeem-ur-Rehman, “Discrete Mathematical Structures”, Pearson Education.
3. D. S. Malik and M. K. Sen, “Discrete Mathematical Structures”, Thompson.

References:

1. C. L. Liu, D. P. Mohapatra, “Elements of Discrete Mathematics” Tata McGrawHill.
2. J. P. Trembley, R. Manohar “Discrete Mathematical Structures with Applications to Computer Science”, Tata Mcgraw-Hill.
3. Y N Singh, “Discrete Mathematical Structures”, Wiley-India.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC306	Electronic Circuits and Communication Fundamentals	05

Course Objectives:

1. To develop the knowledge of semiconductor devices and circuits, and explain their use in communication applications.
2. To inculcate circuit analysis capabilities in students.
3. To make students aware of various types of integrated circuits that can be used in computer applications.
4. To make students aware that knowledge gained in electronic devices and circuits is useful in real life applications.

Course Outcomes:

1. Ability to understand and use semiconductor devices in circuits.
2. Ability to analyze the given circuit.
3. Ability to understand field effect devices and carry out their DC analysis.
4. Ability to understand concept of feedback and oscillations.
5. Ability to use oscillators in various applications.
6. Ability to use operational amplifier in various applications.
7. Ability to understand concept of phase lock loop and their use communication applications.
8. Ability to understand fundamental concepts of communication.
9. Ability to apply knowledge of electronic devices and circuits to communication applications.

Module	Detailed content	Hours
01	**Electronic Circuits** • Field effect based devices and circuits: Junction Field Effect Transistors, JFET Characteristics, FET amplification and switching, DC load line and bias point, ate bias, self bias, voltage divider bias, coupling, bypassing and AC load lines, FET models and parameters, Common source circuit analysis principle of oscillation, FET based Hartley and Colpitts Oscillator. • Crystal oscillator • BJT as power amplifier (only class A and C)	12
02	• Operational Amplifier and its applications: Op-amp parameters and characteristics, Inverting and Non-inverting amplifier, Comparator, Summing Amplifier, Integrator, Differentiator, Zero Crossing Detector. • Phase Lock Loop: Operating principle of PLL, Lock range and capture range.	06
03	**Modulation** • Principles of Analog Communication: Elements of analog communication systems, Theory of amplitude modulation and types of AM, Generation of DSB SC using balanced modulator, Generation of SSB using phase shift method Theory of FM and PM, Generation of FM by Armstrong method	12

04	**Demodulation :** • Principle of super heterodyne receiver. • Foster seely detector for FM detection • Application of PLL (IC 565) as FM detector , Frequency translator, Phase shifter, and freq synthesizer	06
05	• Concept of sampling :Sampling Theorem, Types of sampling Quantization , A/D and D/A conversion concept • Pulse Modulation: generation and detection of PAM, PPM, PWM, PCM, DM and ADM.Principle of TDM and FDM.	12

Text Books:

1. David Bell, 'Electronic Devices and Circuits', Oxford, 5th Edition.
2. Wayne Tomasi 'Electronic Communication Systems (fundamentals through advanced)', Pearson Education, 4th Edition.
3. Ramakant A. Gayakwad, 'Op-amp and linear integrated circuits', PHI, 3rd edition.
4. G. Kennedy, B. Davis, S R M Prasanna, 'Electronic Communication Systems', Mc Graw Hill, 5th Edition.

References:

1. Robert Diffenderfer, 'Electronic Devices: Systems & Applications', Cengage Learning, India Edition.
2. K. R. Botkar, 'Integrated Circuits', Khanna Publishers, 9th Edition
3. Donald Neamen, 'Electronic Circuit Analysis and Design', Tata McGraw Hill,2nd Edition.
4. David Bell, 'Electronic Devices and Circuits', Oxford, 5th Edition.
5. Wayne Tomasi 'Electronic Communication Systems (fundamentals through advanced)', Pearson Education, 4th Edition.
6. Ramakant A. Gayakwad, 'Op-amp and linear integrated circuits', PHI, 3rd edition.
7. G. Kennedy, B. Davis, S R M Prasanna, 'Electronic Communication Systems', Mc Graw Hill, 5th Edition.
8. Robert Diffenderfer, 'Electronic Devices: Systems & Applications', Cengage Learning, India Edition.
9. K. R. Botkar, 'Integrated Circuits', Khanna Publishers, 9th Edition
10. Donald Neamen, 'Electronic Circuit Analysis and Design', Tata McGraw Hill,2nd Edition.

Termwork:

Term work should consist of at least 08 experiments.
Journal must include at least 2 assignments.
The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 5 Marks (Assignment) + 5 (Attendance (theory+practical))
Oral exam will be based on the above syllabus.

Suggested List of Experiments:

1. Study of various test and measuring instruments
2. Implementation of diode detector
3. Implementation of single stage FET amplifier
4. Implementation of oscillators
5. Implementation of IC 741 based application
6. Implementation of IC741 based active filters
7. Implementation of IC555 based application
8. Troubleshooting of given faults
9. Modulation and demodulation of AM/SSB/FM
10. Study of superheterodyne receiver
11. Generation and detection of PAM/PPM/PWM
12. Generation and detection of PCM/DM/ADM
13. Study of FDM and TDM
14. SPICE based simulations

Important Note:

- **50% experiments from communication and 50% experiments from electronic circuits should be taken.**
- **In theory exam the weightage for marks out of 80 : 35 for Devices and 45 for communications**

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC401	Applied Mathaematics IV *	05

Course Objectives:

This course will present matrix theory, Similar matrices and it's application to find the matrics function. Present methods of computing and using eigen values and eigen vectors. Set up and directly evaluate contour integrals Cauchys integral theorem and formula in basic and extended form. Present Taylor and Laurents series to find singularities zero's and poles also presents residues theory and it's applications. Present theory of probability, Baye's Theorem, Expectation and Moments and it's application. Present probability distribution such as binomial, Poisson and normal distribution with their properties. Present sampling theory and it's application for small and large sample. Present methods of computing optimization using simplex method.

Course Outcomes:

Students in this course will apply the method of solving complex integration and computing residues. Use residues to evaluate various contour integrals. Demonstrate ability to manipulate matrices and compute eigen values and eigenvectors.

Students in this course will apply the Procedure and methods to solve technical problems.

Module		
01	**Complex Integration** 1.1 Complex Integration – Line Integral, Cauchy's Integral theorem for simply connected regions, Cauchy's Integral formula(without proof) 1.2 Taylor's and Laurent's series (without proof) 1.3 Zeros, poles of f(z), Residues, Cauchy's Residue theorem 1.4 Applications of Residue theorem to evaluate Integrals of the type $\int_0^{2\Pi} f(\sin\theta, \cos\theta)\,d\theta$, $\int_{-\infty}^{\infty} f(x)\,dx$.	**(10)**
02	**Matrices:-** 2.1 Eigen values and eigen vectors 2.2 Cayley-Hamilton theorem(without proof) 2.3 Similar matrices, diagonalisable of matrix. 2.4 Derogatory and non-derogatory matrices ,functions of square matrix.	**(08)**
03	**Correlation** 3.1Scattered diagrams, Karl Pearson's coefficient of correlation, covariance,	**(04)**

	Spearman's Rank correlation. 3.2 Regression Lines.	
04	**Probability** 4.1 Baye's Theorem, 4.2 Random Variables:- discrete & continuous random variables, expectation, Variance, Probability Density Function & Cumulative Density Function. 4.3 Moments, Moment Generating Function. 4.4 Probability distribution: binomial distribution, Poisson & normal distribution. (For detail study)	**(10)**
05	**Sampling theory** 5.1 Test of Hypothesis, Level of significance, Critical region, One Tailed and two Tailed test, Test of significant for Large Samples:-Means of the samples and test of significant of means of two large samples. 5.2 Test of significant of small samples:- Students t- distribution for dependent and independent samples. 5.3 Chi square test:- Test of goodness of fit and independence of attributes, Contingency table.	**(08)**
06	**Mathematical Programming** 6.1 Types of solution, Standard and Canonical form of LPP, Basic and feasible solutions, simplex method. 6.2 Artificial variables, Big –M method (method of penalty). 6.3 Duality, Dual simplex method. 6.4 Non Linear Programming:-Problems with equality constrains and inequality constrains (No formulation, No Graphical method).	**(08)**

Term work:

Term work shall consist of minimum four SCILAB practicals and six tutorials.

SCILAB practicals	:	10 marks
Tutorials	:	10 marks
Attendance	:	05 marks
Total	**:**	**25 marks**

Text Books:

1. Higher Engineering Mathematics by Grewal B. S. 38th edition, Khanna Publication 2005.
2. Operation Research by Hira & Gupta,S Chand.
3. A Text Book of Applied Mathematics Vol. I & II by P.N.Wartilar &
4. J.N.Wartikar, Pune, Vidyarthi Griha Prakashan., Pune.
5. Probability and Statistics for Engineering, Dr. J Ravichandran, Wiley-India.

Reference Books:

1. Probability & Statistics with reliability by Kishor s. Trivedi, Wiley India.
2. Advanced Engg. Mathematics by C. Ray Wylie & Louis Barrett.TMH International Edition.
3. Mathematical Methods of Science and Engineering by Kanti B. Datta, Cengage Learning.
4. Advanced Engineering Mathematics by Kreyszig E. 9th edition, John Wiley.
5. Operations Research by S.D. Sharma Kedar Nath, Ram Nath & Co. Meerat.
6. Engineering optimization (Theory and Practice) by Singiresu S.Rao, New Age International publication.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC402	Analysis of Algorithm	5

Prequisites : Students should be familiar with Data structure concepts , Discrete structures

Course Objectives:

1. To teach various problem solving strategies..
2. To teach mathematical background for algorithm analysis and implementation of various strategies like divide and conquer, Greedy method, Dynamic programming , Backtracking , branch and bound
3. To teach different string matching algorithms.

Course Outcomes:

1. Ability to select appropriate problem solving strategies.
2. Ability to calculate time complexity and space complexity of an algorithm.
3. Ability to analyze different divide and conquer problems.
4. Ability to analyze different greedy method problems.
5. Ability to analyze different dynamic programming problems.
6. Ability to analyze different backtracking problems.
7. Ability to analyze different string matching algorithms.

Module	Detailed Content	Hrs.
1	**Introduction to analysis of algorithm** • Decision and analysis fundamentals • Performance analysis , space and time complexity • Growth of function – Big –Oh ,Omega , Theta notation • Mathematical background for algorithm analysis • Analysis of selection sort , insertion sort • Randomized algorithms • Recursive algorithms • The substitution method • Recursion tree method • - Master method	**11**
2	**Divide and Conquer** • General method • Binary search • Finding minimum and maximum • Merge sort analysis • Quick sort analysis • Strassen's matrix multiplication • The problem of multiplying long integers	**07**

	• - constructing Tennis tournament	
3	**Greedy Method** • General Method • Knapsack problem • Job sequencing with deadlines • Minimum cost spanning trees-Kruskal and prim's algorithm • Optimal storage on tapes • Single source shortest path	**07**
4	**Dynamic Programming** • General Method • Multistage graphs • all pair shortest path • single source shortest path • Optimal binary search tree • 0/1 knapsack • Travelling salesman problem • - Flow shop scheduling	**08**
5	**Backtracking** • General Method • 8 queen problem(N-queen problem) • Sum of subsets • Graph coloring	**05**
6	**String Matching Algorithms** • The naïve string matching Algorithms • The Rabin Karp algorithm • String matching with finite automata • The knuth-Morris-Pratt algorithm • Longest common subsequence algorithm	**06**
7	**Branch and bound** • General method • 15 puzzle problem • Travelling salesman problem	**04**

Text Books:

1. Ellis horowitz , sartaj Sahni , s. Rajsekaran. "Fundamentals of computer algorithms" University Press.
2. T.H.coreman , C.E. Leiserson,R.L. Rivest, and C. Stein, "Introduction to algorithms", 2nd edition , PHI publication 2005.
3. Alfred v. Aho , John E. Hopcroft , Jeffrey D. Ullman , "Data structures and Algorithm" Pearson education , fourth impression 2009

Reference books:

1. Michael Gooddrich & Roberto Tammassia, "Algorithm design foundation, analysis and internet examples", Second edition , wiley student edition.

Suggested Practicals:

Implementations Programing Language must be in 'C' only.

Module no	Module name	Suggested Experiment list
1	**Introduction to analysis of algorithm:**	selection sort insertion sort (for this experiment comparative analysis on the basis of comparison required to sort list is expected for large values of n)
2	**Divide and Conquer**	-binary search -finding minimum and maximum **-Merge sort analysis*** **-Quick sort analysis*** (the above two experiments marked as * should be considered as single experiment. For this experiment comparative analysis on the basis of comparisons required to sort list is expected for large values of n) -Strassen's matrix multiplication -The problem of multiplying long integers **-constructing Tennis tournament***
3	**Greedy Method**	**-Knapsack problem*** -Job sequencing with deadlines **-Minimum cost spanning trees-Kruskal and prim's algorithm*** -Optimal storage on tapes -Single source shortest path
4	**Dynamic Programming**	-Multistage graphs -all pair shortest path -single source shortest path **-Optimal binary search tree*** -0/1 knapsack **-Travelling salesman problem*** -Flow shop scheduling
5	**Backtracking**	**-8 queen problem(N-queen problem)*** -Sum of subsets -Graph coloring -Knapsack problem
6	**String Matching Algorithms**	-The naïve string matching Algorithms -The Rabin Karp algorithm -String matching with finite automata -The knuth-Morris-Pratt algorithm **-Longest common subsequence algorithm***
7	**Branch and bound**	**-15 puzzle problem*** -Travelling salesman problem

Termwork:

Total experiments to be performed are 12 = (9 + 3) 9 Experiments marked * are mandatory.

For additional 3 experiments teacher can choose experiments from **suggested list.**

The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
Termwork: 25 Marks (total marks) = 15 Marks Experiments + 05 Marks Assignment + 5 (Attendance (theory+practical))

Practical Exam will be based on above syllabus

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC403	Computer Organization and Architecture*	05

Course Objectives:

1. To conceptualize the basics of organizational and architectural issues of a digital computer.
2. To analyze performance issues in processor and memory design of a digital computer.
3. To understand various data transfer techniques in digital computer.
4. To analyze processor performance improvement using instruction level parallelism

Course Outcomes:

1. Ability to understand basic structure of computer.
2. Ability to perform computer arithmetic operations.
3. Ability to understand control unit operations.
4. Ability to design memory organization that uses banks for different word size operations.
5. Ability to understand the concept of cache mapping techniques.
6. Ability to understand the concept of I/O organization.
7. Ability to conceptualize instruction level parallelism.

Pre-requistes: Fundamentals of Computer, Digital Logic Circuits, Programming Languages (C, C++, Java)

Module	Detailed Contents	Hours
1	**Overview of Computer Architecture & Organization:** • Introduction of Computer Organization and Architecture. • Basic organization of computer and block level description of the functional units. • Evolution of Computers, Von Neumann model. • Performance measure of Computer Architecture. • Introduction to buses and connecting I/O devices to CPU and Memory, bus structure.	04
2	**Data Representation and Arithmetic Algorithms:** • Number representation: Binary Data representation, two's complement representation and Floating-point representation. IEEE 754 floating point number representation. • Integer Data computation: Addition, Subtraction. Multiplication: Signed multiplication, Booth's algorithm.	10

	• Division of integers: Restoring and non-restoring division • Floating point arithmetic: Addition, subtraction	
3	**Processor Organization and Architecture:** • CPU Architecture, Register Organization , Instruction formats, basic instruction cycle. Instruction interpretation and sequencing. • Control Unit: Soft wired (Micro-programmed) and hardwired control unit design methods. Microinstruction sequencing and execution. Micro operations, concepts of nano programming. • Introduction to RISC and CISC architectures and design issues. • Case study on 8085 microprocessor: Features, architecture, pin configuration and addressing modes.	12
4	**Memory Organization:** • Introduction to Memory and Memory parameters. Classifications of primary and secondary memories. Types of RAM and ROM, Allocation policies, Memory hierarchy and characteristics. • Cache memory: Concept, architecture (L1, L2, L3), mapping techniques. Cache Coherency, Interleaved and Associative memory. • Virtual Memory: Concept, Segmentation and Paging , Page replacement policies.	12
5	**I/O Organization and Peripherals:** • Input/output systems, I/O modules and 8089 IO processor. • Types of data transfer techniques: Programmed I/O, Interrupt driven I/O and DMA. • Peripheral Devices: Introduction to peripheral devices, scanner, plotter, joysticks, touch pad.	6
6	**Introduction to parallel processing systems:** • Introduction to parallel processing concepts • Flynn's classifications • pipeline processing • instruction pipelining, • pipeline stages • pipeline hazards.	4

Text Books:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, Tata McGraw-Hill.
2. John P. Hayes, "Computer Architecture and Organization", Third Edition.
3. William Stallings, "Computer Organization and Architecture: Designing for Performance", Eighth Edition, Pearson.
4. B. Govindarajulu, "Computer Architecture and Organization: Design Principles and Applications", Second Edition, Tata McGraw-Hill.

Reference Books:

1. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization",First Edition, Wiley-India.
2. "Computer Organization" by ISRD Group, Tata McGraw-Hill.
3. Ramesh Gaonkar, "Microprocessor Architecture, Programming and Applications with the 8085, Fifth Edition,Penram.

Termwork:

Term work should consist of at least 08 experiments.
Journal must include at least 2 assignments.
The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 5 Marks (Assignment) + 5 (Attendance (theory+practical))

oral exam will be based on the above syllabus.

Note:
1. The faculty should conduct eight programming practical / experiments based on the above syllabus including two case studies on recent developments covering the above contents.
All the programs should be implemented in C/C++/Java under Windows or Linux environment.
Experiments can also be conducted using available open source tools.
2. 8085 microprocessor should be included only as a sample case study to visualize the concepts. No questions in University Exams / Class Tests should be asked on 8085 microprocessor.

SUGGESTED LIST OF COA PRACTICAL / EXPERIMENTS

1. To study Full Adder (7483).

2. To study ALU (74181).

3. To study MASM (Micro Assembler).

4. A program for hexadecimal addition and multiplication.

5. A program for binary multiplication.

6. A program for Hamming code generation , detection and correction.

7. A program for Booth's multiplication

8. A program for LRU page replacement algorithm.

9. A program for FIFO page replacement algorithm.

10. A program to simulate the mapping techniques of Cache memory.

    10.1 Direct Mapped cache
    10.2 Associative Mapped cache
    10.3 Set Associative Mapped cache

11. A program to simulate memory allocation policies.

    11.1 First-fit algorithm
    11.2 Best-fit algorithm

12. A program to implement serial communication (PC - PC communication).

13. A program to implement parallel communication. (PC - Printer communication).

14. A program for printer simulation.

15. A program for keyboard simulation.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CEC404	Database Management System	05

Course Objectives:

1. Learn and practice data modeling using the entity-relationship and developing database designs.
2. Understand the use of Structured Query Language (SQL) and learn SQL syntax.
3. Apply normalization techniques to normalize the database
4. Understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.

Course Outcomes:

1. The learner will be able:
2. To describe data models and schemas in DBMS
3. To understand the features of database management systems and Relational database.
4. To use SQL- the standard language of relational databases.
5. To understand the functional dependencies and design of the database.
6. To understand the concept of Transaction and Query processing.

Module	Detailed content	Hours
1	**Introduction Database Concepts:** Introduction, Characteristics of databases, File system V/s Database system, Users of Database system, Concerns when using an enterprise database, Data Independence, DBMS system architecture, Database Administrator,	04
2	**Entity–Relationship Data Model :** Introduction, Benefits of Data Modeling, Types of Models, Phases of Database Modeling, The Entity-Relationship (ER) Model, Generalization, Specialization and Aggregation, Extended Entity-Relationship (EER) Model.	04
3	**Relational Model and Algebra :** Introduction , Mapping the ER and EER Model to the Relational Model , Data Manipulation , Data Integrity ,Advantages of the Relational Model, Relational Algebra , Relational Algebra Queries, Relational Calculus.	08
4	**Structured Query Language (SQL) :** Overview of SQL , Data Definition Commands, Set operations , aggregate function , null values, , Data Manipulation commands, Data Control commands , Views in SQL, Nested	09

	and complex queries .	
5	**Integrity and Security in Database:** Domain Constraints, Referential integrity, Assertions, Trigger, Security, and authorization in SQL	04
6	**Relational–Database Design :** Design guidelines for relational schema, Function dependencies, Normal Forms- 1NF, 2 NF, 3NF, BCNF and 4NF	06
7	**Transactions Management and Concurrency:** Transaction concept, Transaction states, ACID properties, Implementation of atomicity and durability, Concurrent Executions, Serializability, Recoverability, Implementation of isolation, Concurrency Control: Lock-based , Timestamp-based , Validation-based protocols, Deadlock handling, Recovery System: Failure Classification, Storage structure, Recovery & atomicity, Log based recovery, Shadow paging.	08
8	**Query Processing and Optimization: Overview** ,Issues in Query Optimization ,Steps in Query Processing , System Catalog or Metadata, Query Parsing , Query Optimization, Access Paths , Query Code Generation , Query Execution , Algorithms for Computing Selection and Projection , Algorithms for Computing a Join , Computing Aggregation Functions , Cost Based Query Optimization .	05

Text Books:

1. G. K. Gupta :"Database Management Systems", McGraw – Hill.
2. Korth, Slberchatz,Sudarshan, :"Database System Concepts", 6th Edition, McGraw – Hill
3. Elmasri and Navathe, " Fundamentals of Database Systems", 5thEdition, PEARSON Education.
4. Peter Rob and Carlos Coronel, " Database Systems Design, Implementation and Management", Thomson Learning, 5th Edition.

Reference Books :

1. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g,Black Book, Dreamtech Press Mark L. Gillenson, Paulraj Ponniah, " Introduction to Database Management",Wiley
2. Sharaman Shah ,"Oracle for Professional", SPD.
3. Raghu Ramkrishnan and Johannes Gehrke, " Database Management Systems",TMH
4. Debabrata Sahoo "Database Management Systems" Tata McGraw Hill, Schaum's Outline

Termwork:

Term work should consist of at least 12 experiments.
Journal must include at least 2 assignments.
The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 5 Marks (Assignment) + 5 (Attendance (theory+practical))
practical exam will be based on the above syllabus.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.

3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC405	Theoretical Computer Science	4

Course Objectives:

1. Conceptual understanding of fundamentals of Grammars and languages.
2. Build concepts of theoretical design of basic machine, deterministic and non deterministic machines and pushdown machines.
3. Develop understanding of different types of Turing machines and their use.
4. Understand the concept of Undecidability.

Course Outcomes:

1. Understanding of Power and Limitations of theoretical models of Computation.
2. Ability to compare different types of languages and machines.
3. Ability to match constraints of a language to power of machines.

Module	Detailed content	Hours
01	**Introduction:** • Alphabets, Strings and Languages • Chomskey hierarchy and Grammars. • Finite Automata (FA) and Finite State machine (FSM).	03
02	**Regular Grammar (RG):** • Regular Grammar and Regular Expression (RE): Definition, Equivalence and Conversion from RE to RG and RG to RE. • Equivalence of RG and FA, Converting RG to FA and FA to RG. • Equivalence of RE and FA, Converting RE to FA and FA to RE.	04
03	**Finite Automata:** • Deterministic and Nondeterministic Finite Automata (DFA and NFA): Definitions, Languages, Transitions (Diagrams, Functions and Tables). • Eliminating epsilon-transitions from NFA.	05

	• DFA, NFA: Reductions and Equivalence. • FSM with output: Moore and Mealy machines.	
04	**Regular Language (RL):** • Decision properties: Emptiness, Finiteness and Membership. • Pumping lemma for regular languages and its applications. • Closure properties. • Myhill-Nerode Theorem and An application: Text Search.	**04**
05	**Context Free Grammars (CFG):** • Definition, Sentential forms, Leftmost and Rightmost derivations. • Context Free languages (CFL): Parsing and Ambiguity. • CFLs: Simplification and Applications. • Normal Forms: CNF and GNF. • Pumping lemma for CFLs and its applications. • Closure properties and Kleene's closure.	**06**
06	**Pushdown Automata(PDA):** • Definition, Transitions (Diagrams, Functions and Tables), Graphical Notation and Instantaneous Descriptions. • Language of PDA, Pushdown Stack Machine (PSM) as a machine with stack, Start and Final state of PSM. • PDA/PSM as generator, decider and acceptor of CFG • Deterministic PDA (DPDA) and Multi-stack DPDA.	**08**
07	**Turing Machine (TM):** • Definition, Transitions (Diagrams, Functions and Tables). • Design of TM as generator, decider and acceptor. • Variants of TM: Multitrack, Multitape and Universal TM. • Equivalence of Single and Multi Tape TMs. • Power and Limitations of TMs. • Design of Single and Multi Tape TMs as a computer of simple functions: Unary, Binary (Logical and Arithmetic), String operations (Length, Concat, Match, Substring Check, etc)	**10**

08	**Undecidability and Recursively Enumerable Languages:** • Recursive and Recursively Enumerable Languages. • Properties of Recursive and Recursively Enumerable Languages. • Decidability and Undecidability, Halting Problem, Rice's Theorem, Grebach's Theorem, Post Correspondence Problem, • Context Sensitivity and Linear Bound Automata.	**06**
09	**Comparison of scope of languages and machines:** • Subset and Superset relation between FSM, PSM and TM. • Subset and Superset relation between RL, CFL and Context Sensitive Language.	**02**

Text Books:

1. Michael Sipser, " Theory of Computation", Cengage learning.
2. John E. Hopcroft, Rajeev Motwani, Jeffery D. Ullman, " Introduction to Automata Theory, Languages and Computation", Pearson Education

References:

1. J. C. Martin, "Introduction to Languages and the Theory of Computation", Tata McGrawHill.
2. Krishnamurthy E. V., "Introductory Theory of Computer Science", East-West Press.
3. Kavi Mahesh, "Theory of Computation: A Problem Solving Approach", Wiley-India.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Subject Code	Subject Name	Credits
CSC406	Computer Graphics	04

Course Educational Objectives:

1. The main objective is to introduce to the students in the graphics mode, with the help of basic algorithms and methodologies .
2. The objective of the course is to equip students with fundamental knowledge and basic technical competence in the field of computer graphics.
3. Provide an understanding of how a computer draws the fundamental graphics primitives.
4. To learn Computer Graphics methodologies/Algorithms and techniques .
5. To learn Implementation of Computer Graphics Algorithms

Course Outcomes:

Upon successfully completing Fundamentals of Computer graphics course, students will have, at a minimum, the qualities listed in the expected learning outcomes below.

1. Student will have understood basic concepts of computer graphics
2. Acquire knowledge about drawing basic shapes such as lines, circle ellipse, polygon.
3. Shall be able to perform processing of basic shapes by various processing algorithms /techniques.
4. Acquire knowledge about two and three dimensional transformations.
5. Shall be able to apply the transformation algorithms to the basic shapes.
6. Shall have basic knowledge of windowing and clipping.
7. Shall be able to apply various algorithms of clipping.
8. Acquire knowledge about Visible Surface Detection methods
9. Acquire knowledge about Illumination Models and Surface Rendering
10. Acquire knowledge about Color Models

Module	Contents	Hours
1.	**Introduction to Computer Graphics** (a) What is Computer Graphics? (b) Where Computer Generated pictures are used (c) Elements of Pictures created in Computer Graphics (d) Graphics display devices (e) Graphics input primitives and Devices	**(02)**
2.	**Introduction to openGL** (a) Getting started Making pictures	**(02)**

	(b) Drawing basic primitives (c) Simple interaction with mouse and keyboard (**For implementation use openGL programming**)	
3.	**Output Primitives** (a) Points and Lines, Antialiasing (b) Line Drawing algorithms • DDA line drawing algorithm • Bresenham's drawing algorithm • Parallel drawing algorithm (c) Circle and Ellipse generating algorithms • Mid-point Circle algorithm • Mid-point Ellipse algorithm (d) Parametric Cubic Curves • Bezier curves • B-Spline curves	**(06)**
4.	**Filled Area Primitives** (a) Scan line polygon fill algorithm (b) Pattern fill algorithm (c) Inside-Outside Tests (d) Boundary fill algorithms (e) Flood fill algorithms	**(02)**
5.	**2D Geometric Transformations** (a) Basic transformations (b) Matrix representation and Homogeneous Coordinates (c) Composite transformation (d) Other transformations (e) Transformation between coordinated systems	**(04)**
6.	**2D Viewing** (a) Window to Viewport coordinate transformation (b) Clipping operations – Point clipping (c) Line clipping • Cohen – Sutherland line clipping • Liang – Barsky line clipping • Midpoint subdivision (d) Polygon Clipping • Sutherland – Hodgeman polygon clipping • Weiler – Atherton polygon clipping	**(04)**
7.	**3D Geometric Transformations and 3D Viewing** (a) 3D object representation methods B-REP , sweep representations , CSG (b) Basic transformations • Translation • Rotation	**(06)**

	• Scaling (c) Other transformations 1. Reflection 2. Rotation about an arbitrary axis (d) Composite transformations (e) Projections – Parallel and Perspective (f) 3D clipping	
8.	**3D Geometric Transformations and 3D Viewing** (a) Classification of Visible Surface Detection algorithm (b) Back Surface detection method (c) Depth Buffer method (d) Scan line method (e) BSP tree method (f) Area Subdivision method	**(04)**
9.	**Illumination Models and Surface Rendering** (a) Basic Illumination Models (b) Halftone and Dithering techniques (c) Polygon Rendering Constant shading , Goraud Shading , Phong Shading	**(03)**
10.	**11. Fractals** (a) Introduction (b) Fractals and self similarity Successive refinement of curves, Koch curve, Fractional Dimension, (c) String production and peano curves **(For implementation use C Programming)**	**(03)**

The journal should consist of 12 experiments and 3 assignments.

Following is the list of compulsory 10 experiments.

Additional 2 experiments can be implemented relevant to the course

1. Drawing the basic primitives and sierpinsky gasket using openGL*.
2. Create a polyline using mouse interaction using openGL*.
3. Bresenham's line drawing algorithm.
4. Mid-Point ellipse drawing algorithm.
5. Implementing Bezier curve.
6. Scanline fill algorithm.
7. 2D transformations.
8. Any one Line clipping algorithm cohen-sutherland / liang barsky.
9. Polygon Clipping algorithm sutherland hodgeman.
10. Any one Fractal generation (Koch curve / Hilbert curve / peano curves using string production)

***Implementation of experiments 1 and 2 must be in openGL.**
Implementation of experiments 3 to 10 must be done in C language.

Termwork:

The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.

Term Work: 25 Marks (total marks) = 15 Marks (Experiment) + 5 Marks (Assignment) + 5 (Attendance (theory+practical))

Practical Exam will be based on above syllabus

TEXT BOOKS

1. Donald D. Hearn & M. Pauline Baker, " Computer Graphics-C Version", 2nd Edition, Pearson Education, 2002, ISBN 81-7808-794-4
2. F.S.Hill , Jr. , "Computer Graphics using OpenGL" , second edition PHI publication.
3. James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice, 2nd Edition in C, Audison Wesley, ISBN – 981-235-974-5
4. William M. Newman, Roberet F. Sproull, " Principles of Interactive Computer Graphics", Second Edition, Tata McGraw-Hill Edition

REFERENCE BOOKS

1. Rajesh K. Maurya, "Computer Graphics", 1st Edition, Wiley India Publication ISBN 978-81-265-3100-4.
2. Amarendra N Sinha, Arun D Udai, "Computer Graphics" ISBN 10: 0070034378, ISBN 13: 9780070634374, Tata McGraw-Hill Education, 2007.
3. Peter Shirley, Steve Marschner, A K Peters, "Fundamentals of Computer Graphics", 3rd Edition, A. K. Peters Ltd. , Natick, Massachusetts, Distributed by Shroff Publishers and Dist. Pvt. Ltd.
4. Zhigang Xiang, Roy A Plastock, " Computer Graphics", second edition, Shaum's Outlines, Tat McGraw Hill
5. David F. Rogers, "Procedural Elements for Computer Graphics", 2nd Edition, Tata McGraw-Hill Publications, 2001, ISBN 0-07-04-7371-4.

Theory Examination:

1. Question paper will comprise of total 6 questions, each of 20 Marks.
2. Only 4 questions need to be solved.
3. Question 1 will be compulsory and based on maximum part of the syllabus.
4. Remaining questions will be mixed in nature (for example suppose Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.