

UNIVERSITY OF MUMBAI



Bachelor of Engineering

Computer Engineering (Second Year – Sem. III & IV)

Revised course

(REV- 2012) from

Academic Year 2012 -13

Under

FACULTY OF TECHNOLOGY

(As per Semester Based Credit and Grading System)

Preamble

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 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 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					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

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 matrices 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	Complex Integration	
01	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 3.1 Scattered 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

Prerequisites : 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 <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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

	<ul style="list-style-type: none"> • - constructing Tennis tournament 	
3	Greedy Method <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • General Method • 8 queen problem(N-queen problem) • Sum of subsets • - Graph coloring 	05
6	String Matching Algorithms <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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 Programming 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-requisites: Fundamentals of Computer, Digital Logic Circuits, Programming Languages (C, C++, Java)

Module	Detailed Contents	Hours
1	Overview of Computer Architecture & Organization: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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

	<ul style="list-style-type: none"> • Division of integers: Restoring and non-restoring division • Floating point arithmetic: Addition, subtraction 	
3	<p>Processor Organization and Architecture:</p> <ul style="list-style-type: none"> • 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	<p>Memory Organization:</p> <ul style="list-style-type: none"> • 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	<p>I/O Organization and Peripherals:</p> <ul style="list-style-type: none"> • 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	<p>Introduction to parallel processing systems:</p> <ul style="list-style-type: none"> • 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, Silberchatz,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	<p>Introduction:</p> <ul style="list-style-type: none"> • Alphabets, Strings and Languages • Chomsky hierarchy and Grammars. • Finite Automata (FA) and Finite State machine (FSM). 	03
02	<p>Regular Grammar (RG):</p> <ul style="list-style-type: none"> • 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	<p>Finite Automata:</p> <ul style="list-style-type: none"> • Deterministic and Nondeterministic Finite Automata (DFA and NFA): Definitions, Languages, Transitions (Diagrams, Functions and Tables). • Eliminating epsilon-transitions from NFA. 	05

	<ul style="list-style-type: none"> • DFA, NFA: Reductions and Equivalence. • FSM with output: Moore and Mealy machines. 	
04	<p>Regular Language (RL):</p> <ul style="list-style-type: none"> • 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	<p>Context Free Grammars (CFG):</p> <ul style="list-style-type: none"> • 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	<p>Pushdown Automata(PDA):</p> <ul style="list-style-type: none"> • 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	<p>Turing Machine (TM):</p> <ul style="list-style-type: none"> • 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	<p>Undecidability and Recursively Enumerable Languages:</p> <ul style="list-style-type: none"> • 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	<p>Comparison of scope of languages and machines:</p> <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • DDA line drawing algorithm • Bresenham's drawing algorithm • Parallel drawing algorithm (c) Circle and Ellipse generating algorithms <ul style="list-style-type: none"> • Mid-point Circle algorithm • Mid-point Ellipse algorithm (d) Parametric Cubic Curves <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Cohen – Sutherland line clipping • Liang – Barsky line clipping • Midpoint subdivision (d) Polygon Clipping <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Translation • Rotation 	(06)

	<ul style="list-style-type: none"> • Scaling <p>(c) Other transformations</p> <ol style="list-style-type: none"> 1. Reflection 2. Rotation about an arbitrary axis <p>(d) Composite transformations</p> <p>(e) Projections – Parallel and Perspective</p> <p>(f) 3D clipping</p>	
8.	<p>3D Geometric Transformations and 3D Viewing</p> <p>(a) Classification of Visible Surface Detection algorithm</p> <p>(b) Back Surface detection method</p> <p>(c) Depth Buffer method</p> <p>(d) Scan line method</p> <p>(e) BSP tree method</p> <p>(f) Area Subdivision method</p>	(04)
9.	<p>Illumination Models and Surface Rendering</p> <p>(a) Basic Illumination Models</p> <p>(b) Halftone and Dithering techniques</p> <p>(c) Polygon Rendering</p> <p>Constant shading , Gouraud Shading , Phong Shading</p>	(03)
10.	<p>11. Fractals</p> <p>(a) Introduction</p> <p>(b) Fractals and self similarity</p> <p style="padding-left: 40px;">Successive refinement of curves, Koch curve, Fractional Dimension,</p> <p>(c) String production and peano curves</p> <p><u>(For implementation use C Programming)</u></p>	(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.