**AC – 11.05.2017**

**Item No. 4.193**

# UNIVERSITY OF MUMBAI

Revised syllabus (Rev- 2016) from Academic Year 2016 -17

Under

## FACULTY OF TECHNOLOGY

# Computer Engineering

**Second Year** with Effect from **AY 2017-18**
**Third Year** with Effect from **AY 2018-19**
**Final Year** with Effect from **AY 2019-20**

As per **Choice Based Credit and Grading System**
with effect from the AY 2016–17

**Co-ordinator, Faculty of Technology's 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). It is also resolved that course objectives and course outcomes are 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.

Choice 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 2-3 weeks to be utilized for revision, guest lectures, coverage of content beyond syllabus etc.

Choice based Credit and grading system is implemented from the academic year 2016-17 through optional courses at department and institute level. This will be effective for SE, TE and BE from academic year 2017-18, 2018-19 and 2019-20 respectively.

**Dr. S. K. Ukarande**
**Co-ordinator,**
**Faculty of Technology,**
**Member - Academic Council**
**University of Mumbai, Mumbai**

**Chairman’s Preamble:**

Engineering education in India is expanding and is set to increase manifold. The major challenge in the current scenario is to ensure quality to the stakeholders along with 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. Program outcomes are essentially a range of skills and knowledge that a student will have at the time of graduation from the program. In line with this Faculty of Technology of University of Mumbai has taken a lead in incorporating the philosophy of outcome based education in the process of curriculum development.

As the Chairman, Board of Studies in Computer Engineering of the University of Mumbai, I am happy to state here that, the Program Educational Objectives for Undergraduate Program were finalized in a brain storming session, which was attended by more than 85 members from different affiliated Institutes of the University. They are either Heads of Departments or their senior representatives from the Department of Computer Engineering. The Program Educational Objectives finalized for the undergraduate program in Computer Engineering are listed below;

1. To prepare the Learner with a sound foundation in the mathematical, scientific and engineering fundamentals.
2. To motivate the Learner in the art of self-learning and to use modern tools for solving real life problems.
3. To equip the Learner with broad education necessary to understand the impact of Computer Science and Engineering in a global and social context.
4. To encourage, motivate and prepare the Learner’s for Lifelong- learning.
5. To inculcate professional and ethical attitude, good leadership qualities and commitment to social responsibilities in the Learner’s thought process.

In addition to Program Educational Objectives, for each course of the program, objectives and expected outcomes from a learner’s point of view are also included in the curriculum to support the philosophy of outcome based education. I strongly believe that even a small step taken in the right direction will definitely help in providing quality education to the major stakeholders.

**Dr. Subhash K. Shinde**
**Chairman, Board of Studies in Computer Engineering,**
**University of Mumbai, Mumbai.**

## Program Structure B.E. Computer Engineering, (Rev. 2016) w.e.f. AY 2017-18

## S. E. Computer Engineering (Semester-IV)

| Course Code | Course Name | Teaching Scheme (Contact Hours) | | | Credits Assigned | | | |
|---|---|---|---|---|---|---|---|---|
| | | Theory | Pract | Tut | Theory | TW/ Pract | Tut | Total |
| CSC401 | Applied Mathematics- IV | 4+1@ | - | - | 5 | - | - | 5 |
| CSC402 | Analysis of Algorithms | 4 | - | - | 4 | - | - | 4 |
| CSC403 | Computer Organization and Architecture | 4 | - | - | 4 | - | - | 4 |
| CSC404 | Computer Graphics | 4 | - | - | 4 | - | - | 4 |
| CSC405 | Operating System | 4 | - | - | 4 | - | - | 4 |
| CSL401 | Analysis of Algorithms Lab | - | 2 | - | - | 1 | - | 1 |
| CSL402 | Computer Graphics Lab | - | 2 | - | - | 1 | - | 1 |
| CSL403 | Processor Architecture Lab | - | 2 | - | | 1 | - | 1 |
| CSL404 | Operating System Lab | - | 2 | - | - | 1 | - | 1 |
| CSL405 | Open Source Tech Lab | - | 2+2* | - | - | 2 | - | 2 |
| | **Total** | **21** | **12** | - | **21** | **6** | - | **27** |

@ 1 hour to be taken tutorial as class wise .

*2 hours shown as Practical’s to be taken class wise and other 2 hours to be taken as batch wise

| Course Code | Course Name | Examination Scheme | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Theory | | | | | TW | Oral | Oral & Pract | Total |
| | | Internal Assessment | | | End Sem. Exam | Exam Duration ( in Hrs) | | | | |
| | | Test 1 | Test 2 | Avg. | | | | | | |
| CSC401 | Applied Mathematics- IV | 20 | 20 | 20 | 80 | 3 | - | - | - | 100 |
| CSC402 | Analysis of Algorithms | 20 | 20 | 20 | 80 | 3 | - | - | - | 100 |
| CSC403 | Computer Organization and Architecture | 20 | 20 | 20 | 80 | 3 | - | - | - | 100 |
| CSC404 | Computer Graphics | 20 | 20 | 20 | 80 | 3 | - | - | - | 100 |
| CSC405 | Operating System | 20 | 20 | 20 | 80 | 3 | -- | - | - | 100 |
| CSL401 | Analysis of Algorithms Lab | - | - | - | - | - | 25 | -- | 25 | 50 |
| CSL402 | Computer Graphics Lab | - | - | - | - | - | 25 | -- | 25 | 50 |
| CSL403 | Processor Architecture Lab | - | - | - | - | - | 25 | 25 | - | 50 |
| CSL404 | Operating System Lab | - | - | - | - | - | 25 | - | 25 | 50 |
| CSL405 | Open Source Tech Lab | - | - | - | - | - | 25 | --- | 25 | 50 |
| **Total** | | 100 | 100 | 100 | 400 | - | 125 | 25 | 100 | 750 |

| Course Code | Course Name | Credits |
|---|---|---|
| **CSC401** | **Applied Mathematics-IV** | **5** |

**Course Objectives:** The objectives of this course are to teach the students:

1. Matrix theory, and it's application to find the matrix function. Present methods of computing and using Eigen values and Eigen vectors.
2. Set up and directly evaluate contour integrals Cauchy's integral theorem and formula in basic and extended form. Present Taylor and Laurent's series to find singularities zero's and poles also presents residues theory
3. Theory of probability, Baye's Theorem, Expectation and Moments and it's application.
4. Probability distribution such as Binomial, Poisson and Normal distribution with their properties.
5. Sampling theory and it's application for small and large sample and Optimization techniques.

**Course Outcomes:**

1. Students in this course will be able to apply the method of solving complex integration, computing residues & evaluate various contour integrals.
2. Demonstrate ability to manipulate matrices and compute Eigen values and Eigen vectors.
3. Apply the concept of probability distribution to the engineering problems.
4. Apply the concept of sampling theory to the engineering problems.
5. Use matrix algebra with its specific rules to solve the system of linear equation, using concept of Eigen value and Eigen vector to the engineering problems.
6. Apply the concept of Linear & Non-Linear Programming Problem to the engineering problems.

| Module No. | Unit No. | Topics | Hrs. |
|---|---|---|---|
| **1.0** | | **Complex Integration** | **10** |
| | **1.1** | Complex Integration – Line Integral, Cauchy's Integral theorem for simply<br>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 $\int_0^{2\pi} f(\cos\theta,\sin\theta)d\theta$ , $\int_{-\infty}^{\infty} f(x)dx$ | |
| **2.0** | | **Matrices** | **10** |
| | **2.1** | Eigen values and Eigen vectors. | |
| | **2.2** | Cayley-Hamilton theorem(without proof) | |
| | **2.3** | Similar matrices, diagonalisable matrix. | |
| | **2.4** | Derogatory and non-derogatory matrices, Functions of square matrix. | |

|  |  |  |  |
|---|---|---|---|
| **3.0** |  | **Probability** | **10** |
|  | **3.1** | Baye’s Theorem |  |
|  | **3.2** | **Random Variables:** Discrete & continuous random variables, expectation, Variance, Probability Density Function & Cumulative Density Function. |  |
|  | **3.3** | Moments & Moment generating function. |  |
|  | **3.4** | **Probability distribution:** Binomial distribution, Poisson & Normal distribution. (For detail study) |  |
| **4.0** |  | **Sampling Theory (Large Sample test)** | **06** |
|  | **4.1** | Sampling Distribution, Test of Hypothesis, Level of significance, Critical region, One Tailed and Two Tailed test, |  |
|  | **4.2** | Test of significant for Large Samples:-Means of the samples and test of significant of means of two large samples. |  |
| **5.0** |  | **Sampling Theory (Small Sample test)** | **06** |
|  | **5.1** | Test of significant for small samples:- Students t- distribution for dependent and independent samples |  |
|  | **5.2** | Chi square test:- Test of goodness of fit and independence of attributes,Contingency table. |  |
| **6.0** |  | **Mathematical Programming** | **10** |
|  | **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 and Dual simplex method. |  |
|  | **6.4** | Non Linear Programming Problems with equality constrains and inequality Constrains (two or three variables with one constrains) (No formulation, No Graphical method). |  |
|  |  | **Total** | **52** |

**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.

**Assessment:**

**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

**End Semester Theory Examination**:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

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

| Course Code | Course  Name | Credits |
|---|---|---|
| **CSC402** | **Analysis of Algorithms** | **4** |

**Course Objectives:**

4. To provide mathematical approach for Analysis of Algorithms
5. To solve problems using various strategies
6. To analyse strategies for solving problems not solvable in polynomial time.

**Course Outcomes:** At the end of the course student will be able to

1. Analyze the running time and space complexity of algorithms.
2. Describe, apply and analyze the complexity of divide and conquer strategy.
3. Describe, apply and analyze the complexity of greedy strategy.
4. Describe, apply and analyze the complexity of dynamic programming strategy.
5. Explain and apply backtracking, branch and bound and string matching techniques to deal with some hard problems.
6. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.

**Prerequisites:** Students should be familiar with concepts of Data structure and discrete structures.

| Module | Detailed Content | Hours |
|---|---|---|
| 1 | **Introduction to analysis of algorithm**<br>Performance analysis , space and time complexity<br>Growth of function – Big –Oh ,Omega , Theta notation<br>Mathematical background for algorithm analysis,<br>Analysis of selection sort , insertion sort.<br>**Recurrences:**<br>-The substitution method<br>-Recursion tree method<br>-Master method<br>**Divide and Conquer Approach:**<br>General method<br>Analysis of Merge sort, Analysis of Quick sort, Analysis of Binary search, Finding minimum and maximum algorithm and analysis, Strassen's matrix multiplication | 12 |
| 2 | **Dynamic Programming Approach:**<br>General Method<br>Multistage graphs<br>single source shortest path<br>all pair shortest path<br>Assembly-line scheduling<br>0/1 knapsack<br>Travelling salesman problem<br>Longest common subsequence | 08 |
| 3 | **Greedy Method Approach:** | 06 |

| | | |
|---|---|---|
| | General Method<br>Single source shortest path<br>Knapsack problem<br>Job sequencing with deadlines<br>Minimum cost spanning trees-Kruskal and prim’s algorithm<br>Optimal storage on tapes | |
| 4 | **Backtracking and Branch-and-bound:**<br>General Method<br>8 queen problem( N-queen problem)<br>Sum of subsets<br>Graph coloring<br>15 puzzle problem,<br>Travelling salesman problem. | 08 |
| 5 | **String Matching Algorithms:**<br>The naïve string matching Algorithms<br>The Rabin Karp algorithm<br>String matching with finite automata<br>The knuth-Morris-Pratt algorithm | 06 |
| 6 | **Non-deterministic polynomial algorithms:**<br>Polynomial time,<br>Polynomial time verification<br>NP Completeness and reducibility<br>NP Completeness proofs<br>Vertex Cover Problems<br>Clique Problems | 08 |

**Text Books:**

1. T.H.coreman , C.E. Leiserson,R.L. Rivest, and C. Stein, “Introduction to algorithms”, 2nd edition , PHI publication 2005.
2. Ellis horowitz , Sartaj Sahni , S. Rajsekaran. “Fundamentals of computer algorithms” University Press

**Reference Books:**

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, “Algorithms”, Tata McGraw- Hill Edition.
2. S. K. Basu, “Design Methods and Analysis of Algorithm”, PHI.
3. John Kleinberg, Eva Tardos, “Algorithm Design”, Pearson.
4. Michael T. Goodrich, Roberto Tamassia, “Algorithm Design”, Wiley Publication.

## Assessment:

### Internal Assessment:

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

### End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules**..**

| Course Code | Course Name | Credit |
|---|---|---|
| **CSC403** | **Computer Organization and Architecture** | **4** |

**Course Objectives:**

1. To have a thorough understanding of the basic structure and operation of a digital computer.
2. To discuss in detail the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
3. To study the different ways of communicating with I/O devices and standard I/O interfaces.
4. To study the hierarchical memory system including cache memories and virtual memory.

**Course Outcomes:** At the end of the course student should be able-

1. To describe basic structure of the computer system.
2. To demonstrate the arithmetic algorithms for solving ALU operations.
3. To describe instruction level parallelism and hazards in typical processor pipelines.
4. To describe superscalar architectures, multi-core architecture and their advantages
5. To demonstrate the memory mapping techniques.
6. To Identify various types of buses, interrupts and I/O operations in a computer system

**Prerequisite:** Digital Logic Design and Application

| **Sr. No.** | **Module** | **Detailed Content** | **Hours** |
|---|---|---|---|
| 1 | Introduction | Overview of Computer Architecture & Organization<br>• Introduction<br>• Basic organization of computer<br>• Block level description of the functional units.<br>Data Representation and Arithmetic Algorithms:<br>• Integer Data computation: Addition, Subtraction. Multiplication: unsigned multiplication, Booth's algorithm.<br>• Division of integers: Restoring and non restoring division<br>• Floating point representation. IEEE 754 floating point number representation.<br>• Floating point arithmetic: Addition, Subtraction, Multiplication, Division | 08 |
| 2 | Processor Organization and Architecture | • Von Neumann model, Harvard Architecture<br>• Register Organization, Instruction formats, addressing modes, instruction cycle. Instruction interpretation and sequencing.<br>• ALU and Shifters<br>• Basic pipelined datapath and control, Data dependences, data hazards, Branch hazards, delayed branches, branch prediction<br>• Performance measures – CPI, speedup, efficiency, throughput and Amdahl's law | 10 |

| | | | |
|---|---|---|---|
| 3 | Control Unit Design | • Hardwired control unit design methods: State table, delay element, sequence counter with examples like control unit for multiplication and division<br>• Microprogrammed control Unit: Microinstruction sequencing and execution. Micro operations, Wilkie’s microprogrammed Control Unit, Examples on microprograms | 08 |
| 4 | Memory Organization | • Classifications of primary and secondary memories. Types of RAM (SRAM, DRAM, SDRAM, DDR, SSD) and ROM, Characteristics of memory, Memory hierarchy: cost and performance measurement.<br>• Virtual Memory: Concept, Segmentation and Paging, Address translation mechanism.<br>• Interleaved and Associative memory.<br>• Cache memory Concepts, Locality of reference, design problems based on mapping techniques. Cache Coherency, Write Policies | 12 |
| 5 | I/O Organization and Peripherals | • Common I/O device types and characteristics<br>• Types of data transfer techniques: Programmed I/O, Interrupt driven I/O and DMA.<br>• Introduction to buses, Bus arbitration and multiple bus hierarchy<br>• Interrupt types, Interrupts handling | 06 |
| 6 | Advanced Processor Principles | • Introduction to parallel processing, Flynn’s Classification<br>• Concepts of superscalar architecture, out-of-order execution, speculative execution, multithreaded processor, VLIW, data flow computing.<br>• Introduction to Multi-core processor architecture | 08 |

**Text Books:**

1. William Stallings, “Computer Organization and Architecture: Designing for Performance”, Pearson Publication, 10th Edition, 2013
2. John P. Hayes, “Computer Architecture and Organization”, McGraw-Hill, 1988
3. B. Govindarajulu, “Computer Architecture and Organization: Design Principles and Applications”, Second Edition, McGraw-Hill (India),

**Reference Books:**

1. Andrew S. Tanenbaum “Structured Computer Organization”, Pearson, Sixth Edition
2. Morris Mano. “Computer System Architecture” Pearson Publication, 3rd Edition, 2007
3. Kai Hwang, Fayé Alayé Briggs. “Computer architecture and parallel processing”, McGraw-Hill
4. P. Pal Chaudhuri. “Computer Organization and Design” Prentice Hall India, 2004
5. Dr. M. Usha, T.S. Shrikant. “Computer System Architecture and Organization” Wiley India, 2014.

**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

**End Semester Theory Examination**:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

| Course Code | Course  Name | Credits |
|---|---|---|
| **CSC404** | **Computer Graphics** | **4** |

**Course Objectives**

1 To equip students with the fundamental knowledge and basic technical competence in the field of computer graphics.
2 To emphasize on implementation aspect of Computer Graphics Algorithms.
3 To prepare the student for advance areas like Image Processing or Computer Vision or Virtual Reality and professional avenues in the field of Computer Graphics.

**Course Outcomes** : At the end of the course , the students should be able to

1 Understand the basic concepts of Computer Graphics.
2 Demonstrate various algorithms for scan conversion and filling of basic objects and their comparative analysis.
3 Apply geometric transformations, viewing and clipping on graphical objects.
4 Explore solid model representation techniques and projections.
5 Understand visible surface detection techniques and illumination models.

**Prerequisite:** Knowledge of C Programming, Basic Data Structures and Mathematics**.**

| Module No | Detail Syllabus | Hours |
|---|---|---|
| 1 | **Introduction and Overview of Graphics System**:<br>• Definition and Representative uses of computer graphics, classification of application areas, Overview of coordinate systems ,definition of scan conversion, rasterization and rendering.<br>• Raster scan & random scan displays, Flat Panel displays like LCD and LED , architecture of raster graphics system with display processor, architecture of random scan systems. | 03 |
| 2 | **Output Primitives** :<br>• Scan conversions of point,line, circle and ellipse : DDA algorithm and Bresenham algorithm for line drawing, midpoint algorithm for circle, midpoint algorithm for ellipse drawing ( Mathematical derivation for above algorithms is expected )<br>• Aliasing , Antialiasing techniques like Pre and post filtering , super sampling , and pixel phasing ).<br>• Filled Area Primitive: Scan line Polygon Fill algorithm, Inside outside tests, Boundary Fill and Flood fill algorithm. | 12 |
| 3 | **Two Dimensional Geometric Transformations**<br>• Basic transformations : Translation , Scaling , Rotation<br>• Matrix representation and Homogeneous Coordinates<br>• Composite transformation<br>• Other transformations : Reflection and Shear<br>• Raster method for transformation. | 06 |

| | | |
|---|---|---|
| 4 | **Two Dimensional Viewing and Clipping**<br>• Viewing transformation pipeline and Window to Viewport coordinate transformation<br>• Clipping operations – Point clipping , Line clipping algorithms : Cohen – Sutherland , Midpoint subdivision , Liang – Barsky , Polygon Clipping Algorithms : Sutherland – Hodgeman, Weiler – Atherton. | 08 |
| 5 | **Three Dimensional Object Representations , Geometric Transformations and 3D Viewing**<br>• Boundary Representation and Space partitioning representation: Polygon Surfaces , Bezier Curve , Bezier Surface , B-Spline Curve , Sweep Representation, Constructive Solid Geometry ,Octree, Fractal-Geometry : Fractal Dimension, Koch Curve.<br>• 3D Transformations :Translation, Rotation , Scaling and Reflection.<br>• Composite transformations :Rotation about an arbitrary axis<br>• 3D transformation pipeline<br>• Projections – Parallel , Perspective.( Matrix Representation )<br>• 3D clipping. | 12 |
| 6 | **Visible Surface Detection**<br>• Classification of Visible Surface Detection algorithm<br>• Back Surface detection method<br>• Depth Buffer method<br>• Depth Sorting method<br>• Scan line method<br>• Area Subdivision method | 04 |
| 7 | **Illumination Models and Surface Rendering**<br>• Basic Illumination Models : Diffused reflection, Phong Specular reflection Model<br>• Halftone and Dithering techniques<br>• Polygon Rendering :Constant shading , Gouraud Shading , Phong Shading. | 03 |

**Text Books:**

1. “Computer Graphics” C version by Hearn & Baker, 2nd Edition, Pearson
2. “Computer Graphics Principles and Practice in C , 2nd Edition ,James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, Pearson .
3. “Computer Graphics”, by Rajesh K. Maurya, Wiley India Publication.
4. “Computer Graphics “ , by Samit Bhattacharya , Oxford Publication.

**Reference Books:**

1. "Procedural Elements for Computer Graphics " by D. Rogers , Tata McGraw-Hill Publications.
2. "Computer Graphics" , by Zhigang Xiang , Roy Plastock , Schaum's Outlines McGraw-Hill Education
3. "Computer Graphics using OpenGL" , by F.S.Hill , Jr. ,Third edition, Pearson Publications.

## Assessment:

### Internal Assessment:

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

### End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.

2. The students need to solve total 4 questions.

3. Question No.1 will be compulsory and based on entire syllabus.

4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

| Course Code | Course Name | Credits |
|---|---|---|
| **CSC405** | **Operating System** | **4** |

**Course Objectives:**

1. To introduce basic concepts and functions of operating systems.
2. To understand the concept of process, thread and resource management.
3. To understand the concepts of process synchronization and deadlock.
4. To understand various Memory, I/O and File management techniques.

**Course Outcomes:** At the end of the course student should be able to

1. Understand role of Operating System in terms of process, memory, file and I/O management.
2. Apply and analyse the concept of a process, thread, mutual exclusion and deadlock.
3. Evaluate performance of process scheduling algorithms and IPC.
4. Apply and analyse the concepts of memory management techniques.
5. Evaluate the performance of memory allocation and replacement techniques.
6. Apply and analyze different techniques of file and I/O management.

**Prerequisite:** Computer Organization & Architecture

| Sr No | Module | Detailed Content | Hours |
|---|---|---|---|
| 1 | **Operating System Overview** | Operating System Objectives and Functions, The Evolution of Operating Systems, OS Design Considerations for Multiprocessor and Multicore architectures, Operating system structures, System Calls, Linux Kernel and Shell. | 8 hrs |
| 2 | **Process Concept and Scheduling** | **Process:** Concept of a Process, Process States, Process Description, Process Control Block, Operations on Processes.<br>**Threads**: Definition and Types, Concept of Multithreading, Multicore processors and threads.<br>**Scheduling:** Uniprocessor Scheduling - Types of Scheduling: Preemptive and, Non-preemptive, Scheduling Algorithms: FCFS, SJF, SRTN, Priority based, Round Robin, Multilevel Queue scheduling. Introduction to Thread Scheduling, Multiprocessor Scheduling and Linux Scheduling. | 8 hrs |

| | | | |
|---|---|---|---|
| 3 | **Synchronization and Deadlocks** | **Concurrency**: Principles of Concurrency, Inter-Process Communication, Process/Thread Synchronization.<br>**Mutual Exclusion**: Requirements, Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors), Classical synchronization problems: Readers/Writers Problem, Producer and Consumer problem.<br>**Principles of Deadlock**: Conditions and Resource Allocation Graphs, Deadlock Prevention, Deadlock Avoidance: Banker's Algorithm for Single & Multiple Resources, Deadlock Detection and Recovery. Dining Philosophers Problem. | 12 hrs |
| 4 | **Memory Management** | **Memory Management**: Memory Management Requirements, Memory Partitioning: Fixed Partitioning, Dynamic Partitioning, Memory Allocation Strategies: Best-Fit, First Fit, Worst Fit, Next Fit, Buddy System, Relocation. Paging, Segmentation.<br>**Virtual Memory**: Hardware and Control Structures, Demand Paging, Structure of Page Tables, Copy on Write, Page Replacement Strategies: FIFO, Optimal, LRU, LFU, Approximation, Counting Based. Allocation of frames, Thrashing. | 8 hrs |
| 5 | **File Management** | **File Management**: Overview, File Organization and Access, File Directories, File Sharing, Secondary Storage Management, Linux Virtual File System. | 6 hrs |
| 6 | **Input /Output Management** | **I/O Management and Disk Scheduling**: I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling algorithm: FCFS, SSTF, SCAN, CSCAN, LOOK, C-LOOK. Disk Management, Disk Cache, Linux I/O. | 6 hrs |

**Text Books:**

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 8th Edition, 2014, ISBN-10: 0133805913 • ISBN-13: 9780133805918 .
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons , Inc., 9th Edition, 2016, ISBN 978-81-265-5427-0
3. Andrew Tannenbaum, Operating System Design and Implementation, Pearson, 3rd Edition.
4. D.M Dhamdhere, Operating Systems: A Concept Based Approach, Mc-Graw Hill

**Reference Books:**

1. Maurice J. Bach, "Design of UNIX Operating System", PHI
2. Achyut Godbole and Atul Kahate, Operating Systems, Mc Graw Hill Education, 3rd Edition
3. The Linux Kernel Book, Remy Card, Eric Dumas, Frank Mevel, Wiley Publications.

**Assessment:**

**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

**End Semester Theory Examination**:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

| Lab Code | Lab Name | Credit |
|---|---|---|
| **CSL401** | **Analysis of Algorithms Lab** | **1** |

**Lab outcomes:** At the end of the course student will be able to

1. Analyze the complexities of various problems in different domains.
2. Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains.
3. Develop the efficient algorithms for the new problem with suitable designing techniques.
4. Implement the algorithms using different strategies.

**Prerequisites:** Students should be familiar with concepts of Data structure and Discrete structures.

**Description:**

Minimum 2 experiments should be implemented using any language on each algorithm design strategy (Divide and conquer, dynamic programming, Greedy method, backtracking and branch & bound, string matching).

**Suggested Laboratory Experiments:**

| Sr. No. | Module Name | Suggested Experiment List |
|---|---|---|
| 1 | Introduction to analysis of algorithm<br>Divide and Conquer Approach | Selection sort , insertion sort.<br>Merge sort, Quick sort, Binary search. |
| 2 | Dynamic Programming Approach | Multistage graphs,<br>single source shortest path,<br>all pair shortest path,<br>0/1 knapsack,<br>Travelling salesman problem,<br>Longest common subsequence. |
| 3 | Greedy Method Approach | Single source shortest path,<br>Knapsack problem,<br>Job sequencing with deadlines,<br>Minimum cost spanning trees-Kruskal and prim’s algorithm,<br>Optimal storage on tapes. |
| 4 | Backtracking and Branch-and-bound | 8 queen problem ( N-queen problem),<br>Sum of subsets,<br>Graph coloring,<br>15 puzzle problem,<br>Travelling salesman problem. |

| | | |
|---|---|---|
| 5 | String Matching Algorithms | The naïve string matching Algorithms,<br>The Rabin Karp algorithm,<br>String matching with finite automata,<br>The knuth-Morris-Pratt algorithm. |
| 6 | Any two Experiments | This will involve implementation of two algorithms for problems beyond the scope of syllabus.<br>The exact set of algorithms to implement is to be decided by the course instructor. |

**Text Books:**

1. T.H.Coreman , C.E. Leiserson,R.L. Rivest, and C. Stein, “Introduction to algorithms”, 2nd edition , PHI publication 2005.
2. Ellis horowitz , sartaj Sahni , s. Rajsekaran. “Fundamentals of computer algorithms” University Press

**Reference Books:**

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, “Algorithms”, Tata McGraw- Hill Edition.
2. S. K. Basu, “Design Methods and Analysis of Algorithm”, PHI.
3. Dana Vrajittoru and William Knight, “Practical Analysis of Algorithms”, Springer 2014th Edition.

**Term Work:**

Laboratory work must contain implementation of minimum 10 experiments. The final certification and acceptance of term work ensures the satisfactory performance of laboratory work and minimum passing marks in term work. The 25 marks of the term work should be divided as below:
25 Marks (total marks) = 15 Marks Lab. Experiments + 05 Marks Assignments (based on theory syllabus) + 05 (Attendance: theory + practical)

**Oral & Practical Exam** will be based on the experiments implemented in the Laboratory.

| Lab Code | Lab Title | Credit |
|---|---|---|
| **CSL402** | **Computer Graphics Lab** | **1** |

**Lab Objectives**

1 To emphasize on implementation aspect of Computer Graphics Algorithm.

2 To prepare students for advanced areas like Animation, image processing ,virtual reality etc

**Lab Outcomes :** At the end of the course , the students should be able to

1 Explore the working principle, utility of various input/ output devices and graphical tools.

2 Implement various output and filled area primitive algorithms using C/ OpenGL

3 Apply transformation and clipping algorithms on graphical objects.

4 Implementation of curve and fractal generation.

5 Develop a Graphical application based on learned concept.

**Content:**

Scan conversions: lines, circles, ellipses. Filling algorithms, clipping algorithms. 2D and 3D transformation. Curves. Visible surface determination. Simple animations Application of these through exercises in C/C++/ Open GL

**List of Desirable Experiments:**

1. Study and apply basic opengl functions to draw basic primitives. (*)
2. Implement sierpinsky gasket using openGL.
3. Implement DDA Line Drawing algorithms and Bresenham algorithm(*)
4. Implement midpoint Circle algorithm(*)
5. Implement midpoint Ellipse algorithm
6. Implemen tArea Filling Algorithm: Boundary Fill, Flood Fill ,Scan line Polygon Fill (*)
7. Implement Curve : Bezier for n control points , B Spline (Uniform ) ( atleast one)(*)
8. Implement Fractal (Koch Curve)
9. Character Generation : Bit Map method and Stroke Method
10. Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear.(*)
11. Implement Line Clipping Algorithm: Cohen Sutherland / Liang Barsky.(*)
12. Implement polygon clipping algorithm(atleast one)
13. Program to represent a 3D object using polygon surfaces and then perform 3D transformation.
14. Program to perform projection of a 3D object on Projection Plane : Parallel and Perspective.(*)

**Term Work**

1. Term work should consist of at least 12 experiments. (*)→ Practical to be covered necessarily
2. Journal must include at least 2 assignments.
3. Mini Project to perform using C / OpenGL.

   Possible Ideas:

   a. Animation using multiple object

   b. Graphics editor with following features :

      *Draw basic geometrical entities; apply geometrical transformations, Area filling, Clipping against Clip window, displaying the text, displaying bar / line graphs , pie charts etc.

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) = 10 Marks (Experiments)
+ 5 Marks (Mini Project)
+ 5 Marks (Assignments)
+ 5 Marks (Theory + Practical Attendance).

**Oral & Practical exam** will be based on the above content and CSC404: Computer Graphics.

| Lab Code | Lab Name | Credit |
|---|---|---|
| **CSL403** | **Processor Architecture Lab** | **1** |

**Lab Objectives:**

1. To implement the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
2. To study the different ways of communicating with I/O devices and standard I/O interfaces.
3. To design memory subsystem including cache memory
4. To have through understanding of various computer buses

**Lab Outcomes:** At the end of the course student should be

1. Assemble personal computer
2. Design the basic building blocks of a computer: arithmetic-logic unit, registers, central processing unit, and memory.
3. Implement various algorithms like Booth's algorithm for arithmetic operations
4. Describe various I/O buses with merits and demerits.

**Prerequisite:** Digital Logic Design and Applications

**Content:**

| **Sr. No.** | **Module** | **Detailed Content** |
|---|---|---|
| 1 | Overview of Computer Architecture & Organization | • Computer Anatomy- Memory, Ports, Motherboard and add-on cards<br>• Dismantling and assembling PC |
| 2 | Programs on Data Representation and Arithmetic | • Ripple carry adder, Carry look-ahead adder, registers, Multiplication<br>• Booths Algorithm, Restoring and Non restoring Division |
| 3 | Processor Organization and Architecture | • ALU Design, CPU Design<br>• Case Study on multi-core Processors |
| 4 | Memory Organization | • Memory design, Cache Memory design |
| 5 | I/O Organization and Interrupts | • Case study on buses like ISA, PCI, USB etc<br>• Interrupt handling using C/Java Programming |

**Digital Material:**

- **Manual to use the simulator for computer organization and architecture.** Developed by the Department of CSE, IIT kharagpur (http://cse10-iitkgp.virtual-labs.ac.in/ )

**Books:**

1. William Stallings, “Computer Organization and Architecture: Designing for Performance”, Pearson Publication, 10th Edition, 2013
2. B. Govindarajulu, “Computer Architecture and Organization: Design Principles and Applications”, Second Edition, McGraw-Hill (India),
6. Andrew S. Tanenbaum “Structured Computer Organization”, Pearson, Sixth Edition
7. Morris Mano. “Computer System Architecture” Pearson Publication, 3rd Edition, 2007
8. Kai Hwang, Fayé Alayé Briggs. “Computer architecture and parallel processing”, McGraw-Hill
9. P. Pal Chaudhuri. “Computer Organization and Design” Prentice Hall India, 2004
10. Dr. M. Usha, T.S. Shrikant. “Computer System Architecture and Organization” Wiley India, 2014.

**Term Work**

Term work should consist of at least 10-12 experiments and 3-4 assignments based on above content and CSC403: Computer Organization and Architecture

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) = 10 Marks (Experiments)
+ 5 Marks (Mini Project)
+ 5 Marks (Assignments)
+ 5 Marks (Theory + Practical Attendance).

**Oral exam** will be based on the above content and CSC403: Computer Organization and Architecture.

| Lab Code | Lab Name | Credit |
|---|---|---|
| **CSL404** | **Operating System Lab** | **1** |

**Lab Outcome:**

1. Understand basic operating system commands.
2. Understand and explore various system calls.
3. Write shell scripts and shell commands using kernel APIs.
4. Implement and analyze different process scheduling algorithms
5. Implement and analyze different memory management algorithms.
6. Evaluate process management techniques and deadlock handling using simulator.

**Descriptions:**

| Sr. No | Contents |
|---|---|
| 1 | Explore the internal commands of linux like ls,chdir,mkdir,chown,chmod,chgrp,ps etc |
| 2 | Write shell scripts to do the following:<br>➢ Display top 10 processes in descending order<br>➢ Display processes with highest memory usage.<br>➢ Display current logged in user and logname.<br>➢ Display current shell, home directory, operating system type, current path setting, current working directory.<br>➢ Display OS version, release number, kernel version.<br>➢ Illustrate the use of sort, grep, awk, etc. |
| 3 | a) Create a child process in Linux using the fork system call. From the child process obtain the process ID of both child and parent by using getpid and getppid system call. Explore wait and waitpid before termination of process.<br>b) Explore the following system calls: open, read, write, close, getpid, setpid, getuid, getgid, getegid, geteuid. |
| 4 | Implement basic commands of linux like ls, cp, mv and others using kernel APIs. |
| 5 | Write a program to implement any two CPU scheduling algorithms like FCFS, SJF, Round Robin etc. |
| 6 | Write a program to implement dynamic partitioning placement algorithms i.e Best Fit, First-Fit, Worst-Fit etc |
| 7 | Write a program to implement various page replacement policies. |
| 8 | Using the **CPU-OS simulator** analyze and synthesize the following:<br>a. Process Scheduling algorithms.<br>b. Thread creation and synchronization.<br>c. Deadlock prevention and avoidance. |

**Digital Materials:**

1. Download the CPU-OS simulator along with related tutorials from http://www.teach-sim.com
2. http://www.teach-sim.com

**Books:**

1. Linux Kernel Book, by Remy Card, Eric Dumas, Frank Mevel, Wiley India
2. Unix Concepts and Applications, Sumitabha Das, McGraw Hill.

**Term Work:**

- Term work should consist of at least 12 experiments and 2-3 assignments on above content.
- 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) = 10 Marks (Experiments)
  + 5 Marks (Mini Project)
  + 5 Marks (Assignments)
  + 5 Marks (Theory + Practical Attendance).

**Oral & Practical exam** will be based on the above content and CSC405: Operating system syllabus.

| Lab Code | Lab Name | Credit |
|---|---|---|
| **CSL405** | **Open Source Technology Lab** | **2** |

**Course Outcomes:**

1. To understand basic concepts in python and perl.
2. To explore contents of files, directories and text processing with python
3. To develop program for data structure using built in functions in python.
4. To explore django web framework for developing python based web application.
5. To understand file handling and database handling using perl.
6. To explore basics of two way communication between client and server using python and perl

**Prerequisites:** Knowledge of some programming language like C, Java

**Content:**

| Sr. No | Module Name | Detailed Content |
|---|---|---|
| 1 | Python basics | Data types in python ,Operators in python, Input and Output, Control statement, Arrays in python, String and Character in python, Functions, List and Tuples, Dictionaries Exception, Introduction to OOP, Classes , Objects , Interfaces, Inheritance |
| 2 | Advanced Python | Files in Python, Directories, Building Modules, Packages, Text Processing, Regular expression in python. |
| 3 | Data Structure in Python | Link List, Stack, Queues, Dequeues |
| 4 | Python Integration Primer | Graphical User interface ,Networking in Python , Python database connectivity, Introduction to Django |
| 5 | Basics of Perl | Perl Overview, Variables, Control Statements, Subroutines, Objects, Packages and Modules |
| 6 | Perl advanced | Working with Files, Data manipulation, Database Systems, Networking |

**Text Books**

1. Core Python Programming, Dr. R. Nageswara Rao, Dreamtech Press
2. Beginning Python: Using Python 2.6 and Python 3.1. James Payne, Wrox publication
3. Perl: The Complete Reference. Second Edition. Martin C. Brown, McGraw-Hill
4. Introduction to computing and problem solving using python , E Balagurusamy,McGraw Hill Education

**Reference Book**

1. Perl Black Book, 2nd Edition: Steven Holzner,Dreamtech Press
2. Learn Python the Hard Way: (3rd Edition) (Zed Shaw's Hard Way Series)
3. Python Projects , Laura Cassell,Alan Gauld,wrox publication

**Digital Material:**

1. "The Python Tutorial", http://docs.python.org/release/3.0.1/tutorial/
2. Beginning Perl, https://www.perl.org/books/beginning-perl/
3. http://spoken-tutorial.org
4. www.staredusolutions.org

**Suggested experiments using Python:**

1. Exploring basics of python like data types (strings,list,array,dictionaries,set,tuples) and control
2. statements.
3. Creating functions, classes and objects using python. Demonstrate exception handling and inheritance.
4. Exploring Files and directories
   a. Python program to append data to existing file and then display the entire fille
   b. Python program to count number of lines, words and characters in a file.
   c. Python program to display file available in current directory
5. Creating GUI with python containing widgets such as labels, textbox,radio,checkboxes and custom dialog boxes.

6. Menu driven program for data structure using built in function for link list, stack and queues.
7. Program to demonstrate CRUD( **create, read, update and delete)** operations on database (SQLite/ MySQL) using python.
8. Creation of simple socket for basic information exchange between server and client.
9. Creating web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression).

**Suggested experiments using Perl:**

10. Exploring various data type , loops and conditional statement in perl. And Creating functions, packages and modules in perl.
11. Program to demonstrate use of objects and classes in perl.
12. Program to demonstrate file handling, data manipulation and use of regular expression for text processing in perl
13. Program to send email and read content of URL.

**Term Work:**

Students will submit term work in the form of journal that will include:

1. At least 12-14 programs.
2. One mini-project in a group 2-3 student.
3. Two assignments covering whole syllabus.

**Term Work (25)** = 15 marks (Experiments & Assignments)
+ 10 marks (Mini Project)
+ 05 marks (Attendance)

**Practical and oral examination will be based on suggested practical list and entire syllabus.**