UNIVERSITY OF MUMBAI

Bachelor of Engineering

in

- Artificial Intelligence and Data Science
- Artificial Intelligence and Machine Learning
- Cyber Security
- Internet of Things (IoT)
- Data Engineering
- Computer Science and Engineering (Data Science)
- Computer Science and Engineering (Artificial Intelligence and Machine Learning)
- Computer Science and Engineering (Internet of Things and Cyber Security Including Block Chain Technology)

Second Year with Effect from AY 2021-22

Under

FACULTY OF SCIENCE & TECHNOLOGY

(As per AICTE guidelines with effect from the academic year 2019–2020)

UNIVERSITY OF MUMBAI

Syllabus for Approval

Sr. No.	Heading	Particulars
1	Title of the Course	**Second Year Engineering (Eight New Branches)**
2	Eligibility for Admission	**After Passing First Year Engineering as per the Ordinance 0.6242**
3	Passing Marks	**40%**
4	Ordinances / Regulations (if any)	**Ordinance 0.6242**
5	No. of Years / Semesters	**8 semesters**
6	Level	~~P.G.~~ / **U.G.**/ ~~Diploma / Certificate~~ (Strike out which is not applicable)
7	Pattern	~~Yearly~~ / **Semester** (Strike out which is not applicable)
8	Status	**New** / ~~Revised~~ (Strike out which is not applicable)
9	To be implemented from Academic Year	**With effect from Academic Year: 2021-2022**

Dr. S. K. Ukarande
Associate Dean
Faculty of Science and Technology
University of Mumbai

Dr Anuradha Muzumdar
Dean
Faculty of Science and Technology
University of Mumbai

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 Science and Technology (in particular Engineering)of University of Mumbai has taken a lead in incorporating philosophy of outcome based education in the process of curriculum development.

Faculty 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. 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. Credit assignment for courses is based on 15 weeks teaching learning process, however content of courses is to be taught in 13 weeks and remaining 2 weeks to be utilized for revision, guest lectures, coverage of content beyond syllabus etc.

There was a concern that the earlier revised curriculum more focused on providing information and knowledge across various domains of the said program, which led to heavily loading of students in terms of direct contact hours. In this regard, faculty of science and technology resolved that to minimize the burden of contact hours, total credits of entire program will be of 170, wherein focus is not only on providing knowledge but also on building skills, attitude and self learning. Therefore in the present curriculum skill based laboratories and mini projects are made mandatory across all disciplines of engineering in second and third year of programs, which will definitely facilitate self learning of students. The overall credits and approach of curriculum proposed in the present revision is in line with AICTE model curriculum.

The present curriculum will be implemented for Second Year of Engineering from the academic year 2021-22. Subsequently this will be carried forward for Third Year and Final Year Engineering in the academic years 2022-23, 2023-24, respectively.

Dr. S. K. Ukarande
Associate Dean
Faculty of Science and Technology
University of Mumbai

Dr Anuradha Muzumdar
Dean
Faculty of Science and Technology
University of Mumbai

Incorporation and Implementation of Online Contents from NPTEL/ Swayam Platform

The curriculum revision is mainly focused on knowledge component, skill based activities and project based activities. Self learning opportunities are provided to learners. In the revision process this time in particular Revised syllabus of 'C ' scheme wherever possible additional resource links of platforms such as NPTEL, Swayam are appropriately provided. In an earlier revision of curriculum in the year 2012 and 2016 in Revised scheme 'A' and 'B' respectively, efforts were made to use online contents more appropriately as additional learning materials to enhance learning of students.

In the current revision based on the recommendation of AICTE model curriculum overall credits are reduced to 171, to provide opportunity of self learning to learner. Learners are now getting sufficient time for self learning either through online courses or additional projects for enhancing their knowledge and skill sets.

The Principals/ HoD's/ Faculties of all the institute are required to motivate and encourage learners to use additional online resources available on platforms such as NPTEL/ Swayam. Learners can be advised to take up online courses, on successful completion they are required to submit certification for the same. This will definitely help learners to facilitate their enhanced learning based on their interest.

Dr. S. K. Ukarande
Associate Dean
Faculty of Science and Technology
University of Mumbai

Dr Anuradha Muzumdar
Dean
Faculty of Science and Technology
University of Mumbai

Preface by Board of Studies in
Computer Engineering

Dear Students and Teachers, we, the members of Board of Studies Computer Engineering, are very happy to present Second Year Computer Engineering syllabus effective from the Academic Year 2020-21 (REV-2019'C' Scheme). We are sure you will find this syllabus interesting and challenging.

Computer Engineering is one of the most sought-after courses amongst engineering students hence there is a continuous requirement of revision of syllabus. The syllabus focuses on providing a sound theoretical background as well as good practical exposure to students in the relevant areas. It is intended to provide a modern, industry-oriented education in Computer Engineering. It aims at producing trained professionals who can successfully acquainted with the demands of the industry worldwide. They obtain skills and experience in up-to-date the knowledge to analysis, design, implementation, validation, and documentation of computer software and systems.

The revised syllabus falls in line with the objectives of affiliating University, AICTE, UGC, and various accreditation agencies by keeping an eye on the technological developments, innovations, and industry requirements.

The salient features of the revised syllabus are:

1. Reduction in credits to 170 is implemented to ensure that students have more time for extracurricular activities, innovations, and research.
2. Introduction of Skill Based Lab and Mini Project to showcase their talent by doing innovative projects that strengthen their profile and increases the chance of employability.
3. Students are encouraged to take up part of course through MOOCs platform SWAYAM

We would like to place on record our gratefulness to the faculty, students, industry experts and stakeholders for having helped us in the formulation of this syllabus.

Board of Studies in Computer Engineering

Prof. Sunil Bhirud   : Chairman

Prof. Sunita Patil   : Member

Prof. Leena Raga   : Member

Prof. Subhash Shinde   : Member

Prof. Meera Narvekar   : Member

Prof. Suprtim Biswas   : Member

Prof. Sudhir Sawarkar   : Member

Prof. Dayanand Ingle   : Member

Prof. Satish Ket   : Member

Program Structure for Second Year Computer Engineering

UNIVERSITY OF MUMBAI (With Effect from 2020-2021)
Semester III

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
CSC301	Engineering Mathematics-III	3	--	1*	3	--	1	4
CSC302	Discrete Structures and Graph Theory	3	--	--	3	--	--	3
CSC303	Data Structure	3	--	--	3	--	--	3
CSC304	Digital Logic & Computer Architecture	3	--	--	3	--	--	3
CSC305	Computer Graphics	3	--	--	3	--	--	3
CSL301	Data Structure Lab	--	2	--	--	1	--	1
CSL302	Digital Logic & Computer Architecture Lab	--	2	--	--	1	--	1
CSL303	Computer Graphics Lab	--	2	--	--	1	--	1
CSL304	Skill base Lab course: Object Oriented Programming with Java	--	2+2*	--	--	2	--	2
CSM301	Mini Project – 1 A	--	4$	--	--	2	--	2
Total		**15**	**14**	**1**	**15**	**07**	**1**	**23**

Course Code	Course Name	Examination Scheme							
		Theory					Term Work	Pract & oral	Total
		Internal Assessment			End Sem. Exam	Exam. Duration (in Hrs)			
		Test 1	Test 2	Avg					
CSC301	Engineering Mathematics-III	20	20	20	80	3	25	--	125
CSC302	Discrete Structures and Graph Theory	20	20	20	80	3	--	--	100
CSC303	Data Structure	20	20	20	80	3	--	--	100
CSC304	Digital Logic & Computer Architecture	20	20	20	80	3	--	--	100
CSC305	Computer Graphics	20	20	20	80	3	--	--	100
CSL301	Data Structure Lab	--	--	--	--	--	25	25	50
CSL302	Digital Logic & Computer Architecture Lab	--	--	--	--	--	25	--	25
CSL303	Computer Graphics Lab	--	--	--	--	--	25	25	50
CSL304	Skill base Lab course: Object Oriented Programming with Java	--	--	--	--	--	50	25	75
CSM301	Mini Project – 1 A	--	--	--	--	--	25	25	50
Total		**--**	**--**	**100**	**400**	**--**	**175**	**100**	**775**

*Should be conducted batch wise and

$ indicates workload of Learner (Not Faculty), Students can form groups with minimum 2 (Two) and not more than 4 (Four), Faculty Load: 1 hour per week per four groups

Program Structure for Second Year Computer Engineering
UNIVERSITY OF MUMBAI (With Effect from 2020-2021)
Semester IV

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
CSC401	Engineering Mathematics-IV	3	--	1*	3	--	1	4
CSC402	Analysis of Algorithm	3	--	--	3	--	--	3
CSC403	Database Management System	3	--	--	3	--	--	3
CSC404	Operating System	3	--	--	3	--	--	3
CSC405	Microprocessor	3	--	--	3	--	--	3
CSL401	Analysis of Algorithm Lab	--	2	--	--	1	--	1
CSL402	Database Management System Lab	--	2	--	--	1	--	1
CSL403	Operating System Lab	--	2	--	--	1	--	1
CSL404	Microprocessor Lab	--	2	--	--	1	--	1
CSL405	Skill Base Lab Course: Python Programming	--	2*+2	--	--	2	--	2
CSM401	Mini Project 1-B	--	4$	--	--	2	--	2
Total		**15**	**16**	**1**	**15**	**7**	**1**	**24**

Course Code	Course Name	Examination Scheme							
		Theory					Term Work	Pract & oral	Total
		Internal Assessment			End Sem. Exam.	Exam. Duration (in Hrs)			
		Test 1	Test 2	Avg.					
CSC401	Engineering Mathematics - IV	20	20	20	80	3	25	--	125
CSC402	Analysis of Algorithm	20	20	20	80	3	--	--	100
CSC403	Database Management System	20	20	20	80	3	--	--	100
CSC404	Operating System	20	20	20	80	3	--	--	100
CSC405	Microprocessor	20	20	20	80	3	--	--	100
CSL401	Analysis of Algorithm Lab	--	--	--	--	--	25	25	50
CSL402	Database Management System Lab	--	--	--	--	--	25	25	50
CSL403	Operating System Lab	--	--	--	--	--	25	25	50
CSL404	Microprocessor Lab	--	--	--	--	--	25	--	25
CSL405	Skill Base Lab Course: Python Programming	--	--	--	--	--	25	--	25
CSM401	Mini Project 1-B	--	--	--	--	--	25	25	50
Total		**--**	**--**	**100**	**400**	**--**	**175**	**100**	**775**

*Should be conducted batchwise and

$ indicates workload of Learner (Not Faculty), Students can form groups with minimum 2 (Two) and not more than 4 (Four), Faculty Load: 1 hour per week per four groups.

Course Code	Course Name	Credits
CSC301	**Engineering Mathematics-III**	**4**

Pre-requisite: Engineering Mathematics-I, Engineering Mathematics-II
Course Objectives: The course aims:

1	To learn the Laplace Transform, Inverse Laplace Transform of various functions, its applications.
2	To understand the concept of Fourier Series, its complex form and enhance the problem-solving skills.
3	To understand the concept of complex variables, C-R equations with applications.
4	To understand the basic techniques of statistics like correlation, regression, and curve fitting for data analysis, Machine learning, and AI.
5	To understand some advanced topics of probability, random variables with their distributions and expectations.

Course Outcomes: On successful completion, of course, learner/student will be able to:

1	Understand the concept of Laplace transform and its application to solve the real integrals in engineering problems.
2	Understand the concept of inverse Laplace transform of various functions and its applications in engineering problems.
3	Expand the periodic function by using the Fourier series for real-life problems and complex engineering problems.
4	Understand complex variable theory, application of harmonic conjugate to get orthogonal trajectories and analytic functions.
5	Apply the concept of Correlation and Regression to the engineering problems in data science, machine learning, and AI.
6	Understand the concepts of probability and expectation for getting the spread of the data and distribution of probabilities.

Module	Detailed Contents		Hours
1	**Laplace Transform**		7
	1.1	Definition of Laplace transform, Condition of Existence of Laplace transform.	
	1.2	Laplace Transform (L) of standard functions like $\square^{\square\square}$, $(\square\square)$, $\square\square\square(\square\square)$, $\square\square\square h(\square\square)$, $\square\square\square h(\square\square)$ and $\square^{\square}$, $\square \geq 0$.	
	1.3	Properties of Laplace Transform: Linearity, First Shifting Theorem, Second Shifting Theorem, Change of Scale, Multiplication by *t*, Division by *t*, Laplace Transform of derivatives and integrals (Properties without proof).	
	1.4	Evaluation of real improper integrals by using Laplace Transformation.	
	1.5	**Self-learning Topics:** Laplace Transform: Periodic functions, Heaviside's Unit Step function, Dirac Delta Function, Special functions (Error and Bessel)	
2	**Inverse Laplace Transform**		7
	2.1	Definition of Inverse Laplace Transform, Linearity property, Inverse Laplace Transform of standard functions, Inverse Laplace transform using derivatives.	
	2.2	Partial fractions method to find Inverse Laplace transform.	
	2.3	Inverse Laplace transform using Convolution theorem (without proof)	
	2.4	**Self-learning Topics:** Applications to solve initial and boundary value	

		problems involving ordinary differential equations.	
3	**Fourier Series:**		7
	3.1	Dirichlet's conditions, Definition of Fourier series and Parseval's Identity (without proof).	
	3.2	Fourier series of periodic function with period 2π and $2l$.	
	3.3	Fourier series of even and odd functions.	
	3.4	Half range Sine and Cosine Series.	
	3.5	**Self-learning Topics:** Orthogonal and orthonormal set of functions, Complex form of Fourier Series, Fourier Transforms.	
4	**Complex Variables:**		7
	4.1	Function $f(z)$ of complex variable, Limit, Continuity and Differentiability of $f(z)$, Analytic function: Necessary and sufficient conditions for $f(z)$ to be analytic (without proof).	
	4.2	Cauchy-Riemann equations in Cartesian coordinates (without proof).	
	4.3	Milne-Thomson method: Determine analytic function $f(z)$ when real part (u), imaginary part (v) or its combination (u+v / u-v) is given.	
	4.4	Harmonic function, Harmonic conjugate and Orthogonal trajectories.	
	4.5	**Self-learning Topics:** Conformal mapping, Linear and Bilinear mappings, cross ratio, fixed points and standard transformations.	
5	**Statistical Techniques**		6
	5.1	Karl Pearson's coefficient of correlation (r)	
	5.2	Spearman's Rank correlation coefficient (R) (with repeated and non-repeated ranks)	
	5.3	Lines of regression	
	5.4	Fitting of first- and second-degree curves.	
	5.5	**Self-learning Topics:** Covariance, fitting of exponential curve.	
6	**Probability**		6
	6.1	Definition and basics of probability, conditional probability.	
	6.2	Total Probability theorem and Bayes' theorem.	
	6.3	Discrete and continuous random variable with probability distribution and probability density function.	
	6.4	Expectation, Variance, Moment generating function, Raw and central moments up to 4th order.	
	6.5	**Self-learning Topics:** Skewness and Kurtosis of distribution (data).	

References:	
1	Higher Engineering Mathematics, Dr. B. S. Grewal, Khanna Publication.
2	Advanced Engineering Mathematics, Erwin Kreyszig, Wiley Eastern Limited.
3	Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa Publication.
4	Complex Variables and Applications, Brown and Churchill, McGraw-Hill Education.
5	Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill Education.
6	Theory and Problems of Fourier Analysis with applications to BVP, Murray Spiegel, Schaum's Outline Series.

Term Work:	
General Instructions:	
1	Batch wise tutorials have to be conducted. The number of students per batch will be as per University pattern for practical.
2	Students must be encouraged to write at least 6 class tutorials on the entire syllabus.
3	A group of 4-6 students should be assigned a self-learning topic. Students should prepare a presentation/problem solving of 10-15 minutes. This will be considered as a mini project in Engineering Mathematics. This project will be graded out of 10 marks depending on the performance of the students.

The distribution of Term Work marks will be as follows:	
1	Attendance (Theory and Tutorial)	05 marks
2	Class Tutorials on entire syllabus	10 marks
3	Mini project	10 marks

Assessment:

Internal Assessment Test:

The assessment consists of two class tests of 20 marks each. The 1stclass test (Internal Assessment I) has to be conducted when approximately 40% of the syllabus is completed. The 2nd class test has to be conducted (Internal Assessment II) when an additional 35% syllabus is completed. The duration of each test will be for one hour.

End Semester Theory Examination:

1	The question paper will comprise a total of 6 questions, each carrying 20 marks.
2	Out of the 6 questions, 4 questions have to be attempted.
3	Question 1, based on the entire syllabus, will have 4sub-questions of 5 marks each and is compulsory.
4	Question 2 to Question 6 will have 3 sub-questions, each of 6, 6, and 8 marks, respectively.
5	Each sub-question in (4) will be from different modules of the syllabus.
6	Weightage of each module will be proportional to the number of lecture hours, as mentioned in the syllabus.

Course Code	Course Name	Credits
CSC302	**Discrete Structures and Graph Theory**	**3**

Pre-requisite: Basic Mathematics	

Course Objectives: The course aims:

1	Cultivate clear thinking and creative problem solving.
2	Thoroughly train in the construction and understanding of mathematical proofs. Exercise common mathematical arguments and proof strategies.
3	To apply graph theory in solving practical problems.
4	Thoroughly prepare for the mathematical aspects of other Computer Engineering courses

Course Outcomes: On successful completion, of course, learner/student will be able to:

1	Understand the notion of mathematical thinking, mathematical proofs and to apply them in problem solving.
2	Ability to reason logically.
3	Ability to understand relations, functions, Diagraph and Lattice.
4	Ability to understand and apply concepts of graph theory in solving real world problems.
5	Understand use of groups and codes in Encoding-Decoding
6	Analyze a complex computing problem and apply principles of discrete mathematics to identify solutions

Module	Detailed Contents		Hours
1	**Logic**		6
	1.1	Propositional Logic, Predicate Logic, Laws of Logic, Quantifiers, Normal Forms, Inference Theory of Predicate Calculus, Mathematical Induction.	
2	**Relations and Functions**		**6**
	2.1	Basic concepts of Set Theory	
	2.2	**Relations:** Definition, Types of Relations, Representation of Relations, Closures of Relations, Warshall's algorithm, Equivalence relations and Equivalence Classes	
	2.3	**Function**s: Definition, Types of functions, Composition of functions, Identity and Inverse function	
3	**Posets and Lattice**		**5**
	3.1	Partial Order Relations, Poset, Hasse Diagram, Chain and Anti chains, Lattice, Types of Lattice, Sub lattice	
4	**Counting**		**6**
	4.1	Basic Counting Principle-Sum Rule, Product Rule, Inclusion-Exclusion Principle, Pigeonhole Principle	
	4.2	Recurrence relations, Solving recurrence relations	
5	**Algebraic Structures**		**8**
	5.1	**Algebraic structures with one binary operation:** Semi group, Monoid, Groups, Subgroups, Abelian Group, Cyclic group, Isomorphism	
	5.2	**Algebraic structures with two binary operations:** Ring	
	5.3	**Coding Theory**: Coding, binary information and error detection, decoding and error correction	
6	**Graph Theory**		**8**
		Types of graphs, Graph Representation, Sub graphs, Operations on Graphs, Walk, Path, Circuit, Connected Graphs, Disconnected Graph, Components, Homomorphism and Isomorphism of Graphs, Euler and Hamiltonian Graphs, Planar Graph, Cut Set, Cut Vertex,	

		Applications.	

Textbooks:	
1	Bernad Kolman, Robert Busby, Sharon Cutler Ross, Nadeem-ur-Rehman, "Discrete Mathematical Structures", Pearson Education.
2	C. L. Liu "Elements of Discrete Mathematics", second edition 1985, McGraw-Hill Book Company. Reprinted 2000.
3	K. H. Rosen, "Discrete Mathematics and applications", fifth edition 2003, Tata McGraw Hill Publishing Company

References:	
1	Y N Singh, "Discrete Mathematical Structures", Wiley-India.
2	J. L. Mott, A. Kandel, T. P. Baker, "Discrete Mathematics for Computer Scientists and Mathematicians", Second Edition 1986, Prentice Hall of India.
3	J. P. Trembley, R. Manohar "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill Publishing Company
4	Seymour Lipschutz, Marc Lars Lipson, "Discrete Mathematics" Schaum"s Outline, McGraw Hill Education.
5	Narsing Deo, "Graph Theory with applications to engineering and computer science", PHI Publications.
6	P. K. Bisht, H. S. Dhami, "Discrete Mathematics", Oxford press.

Assessment:

Internal Assessment Test:

The assessment consists of two class tests of 20 marks each. The 1st class test (Internal Assessment I) has to be conducted when approximately 40% of the syllabus is completed. The 2nd class test has to be conducted (Internal Assessment II) when an additional 40% syllabus is completed. The duration of each test will be for one hour.

End Semester Theory Examination:

1	The question paper will comprise a total of 6 questions, each carrying 20 marks.
2	Out of the 6 questions, 4 questions have to be attempted.
3	Question 1, based on the entire syllabus, will have 4 sub-questions of 5 marks each and is compulsory.
4	Question 2 to Question 6 will have 3 sub-questions, each of 6, 6, and 8 marks, respectively.
5	Each sub-question in (4) will be from different modules of the syllabus.
6	Weightage of each module will be proportional to the number of lecture hours, as mentioned in the syllabus.

Useful Links	
1	https://www.edx.org/learn/discrete-mathematics
2	https://www.coursera.org/specializations/discrete-mathematics
3	https://nptel.ac.in/courses/106/106/106106094/
4	https://swayam.gov.in/nd1_noc19_cs67/preview

Course Code	Course Name	Credit
CSC303	**Data Structure**	**03**

Pre-requisite: C Programming

Course Objectives: The course aims:

1	To understand the need and significance of Data structures as a computer Professional.
2	To teach concept and implementation of linear and Nonlinear data structures.
3	To analyze various data structures and select the appropriate one to solve a specific real-world problem.
4	To introduce various techniques for representation of the data in the real world.
5	To teach various searching techniques.

Course Outcomes:

1	Students will be able to implement Linear and Non-Linear data structures.
2	Students will be able to handle various operations like searching, insertion, deletion and traversals on various data structures.
3	Students will be able to explain various data structures, related terminologies and its types.
4	Students will be able to choose appropriate data structure and apply it to solve problems in various domains.
5	Students will be able to analyze and Implement appropriate searching techniques for a given problem.
6	Students will be able to demonstrate the ability to analyze, design, apply and use data structures to solve engineering problems and evaluate their solutions.

Module		Detailed Content	Hours
1		**Introduction to Data Structures**	2
	1.1	Introduction to Data Structures, Concept of ADT, Types of Data Structures-Linear and Nonlinear, Operations on Data Structures.	
2		**Stack and Queues**	8
	2.1	Introduction, ADT of Stack, Operations on Stack, Array Implementation of Stack, Applications of Stack-Well form-ness of Parenthesis, Infix to Postfix Conversion and Postfix Evaluation, Recursion.	
	2.2	Introduction, ADT of Queue, Operations on Queue, Array Implementation of Queue, Types of Queue-Circular Queue, Priority Queue, Introduction of Double Ended Queue, Applications of Queue.	
3		**Linked List**	10
	3.1	Introduction, Representation of Linked List, Linked List v/s Array, Types of Linked List - Singly Linked List, Circular Linked List, Doubly Linked List, Operations on Singly Linked List and Doubly Linked List, Stack and Queue using Singly Linked List, Singly Linked List Application-Polynomial Representation and Addition.	
4		**Trees**	11
	4.1	Introduction, Tree Terminologies, Binary Tree, Binary Tree Representation, Types of Binary Tree, Binary Tree Traversals, Binary Search Tree, Operations on Binary Search Tree, Applications of Binary Tree-Expression Tree, Huffman Encoding, Search Trees-AVL, rotations in AVL Tree, operations on AVL Tree, Introduction of B Tree, B+ Tree.	
5		**Graphs**	4

	5.1	Introduction, Graph Terminologies, Representation of Graph, Graph Traversals-Depth First Search (DFS) and Breadth First Search (BFS), Graph Application-Topological Sorting.	
6		**Searching Techniques**	4
	6.1	Linear Search, Binary Search, Hashing-Concept, Hash Functions, Collision resolution Techniques	

Textbooks:	
1	Aaron M Tenenbaum, Yedidyah Langsam, Moshe J Augenstein, "Data Structures Using C", Pearson Publication.
2	Reema Thareja, "Data Structures using C", Oxford Press.
3	Richard F. Gilberg and Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", 2nd Edition, CENGAGE Learning.
4	Jean Paul Tremblay, P. G. Sorenson, "Introduction to Data Structure and Its Applications", McGraw-Hill Higher Education
5	Data Structures Using C, ISRD Group, 2nd Edition, Tata McGraw-Hill.
References:	
1	Prof. P. S. Deshpande, Prof. O. G. Kakde, "C and Data Structures", DreamTech press.
2	E. Balagurusamy, "Data Structure Using C", Tata McGraw-Hill Education India.
3	Rajesh K Shukla, "Data Structures using C and C++", Wiley-India
4	GAV PAI, "Data Structures", Schaum's Outlines.
5	Robert Kruse, C. L. Tondo, Bruce Leung, "Data Structures and Program Design in C", Pearson Edition

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 additional40% syllabus is completed. Duration of each test shall be one hour.

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

Useful Links	
1	https://nptel.ac.in/courses/106/102/106102064/
2	https://www.coursera.org/specializations/data-structures-algorithms
3	https://www.edx.org/course/data-structures-fundamentals
4	https://swayam.gov.in/nd1_noc19_cs67/preview

Course Code	Course Name	Credit
CSC304	Digital Logic & Computer Organization and Architecture	3

Pre-requisite: Knowledge on number systems

Course Objective:

1	To have the rough understanding of the basic structure and operation of basic digital circuits and digital computer.
2	To discuss in detail arithmetic operations in digital system.
3	To discuss generation of control signals and different ways of communication with I/O devices.
4	To study the hierarchical memory and principles of advanced computing.

Course Outcome:

1	To learn different number systems and basic structure of computer system.
2	To demonstrate the arithmetic algorithms.
3	To understand the basic concepts of digital components and processor organization.
4	To understand the generation of control signals of computer.
5	To demonstrate the memory organization.
6	To describe the concepts of parallel processing and different Buses.

Module		Detailed Content	Hours
1		**Computer Fundamentals**	5
	1.1	Introduction to Number System and Codes	
	1.2	Number Systems: Binary, Octal, Decimal, Hexadecimal,	
	1.3	Codes: Grey, BCD, Excess-3, ASCII, Boolean Algebra.	
	1.4	Logic Gates: AND, OR, NOT, NAND, NOR, EX-OR	
	1.5	Overview of computer organization and architecture.	
	1.6	Basic Organization of Computer and Block Level functional Units, Von-Neumann Model.	
2		**Data Representation and Arithmetic algorithms**	8
	2.1	Binary Arithmetic: Addition, Subtraction, Multiplication, Division using Sign Magnitude, 1's and 2's compliment, BCD and Hex Arithmetic Operation.	
	2.2	Booths Multiplication Algorithm, Restoring and Non-restoring Division Algorithm.	
	2.3	IEEE-754 Floating point Representation.	
3		**Processor Organization and Architecture**	6
	3.1	Introduction: Half adder, Full adder, MUX, DMUX, Encoder, Decoder(IC level).	
	3.2	Introduction to Flip Flop: SR, JK, D, T (Truth table).	
	3.3	Register Organization, Instruction Formats, Addressing modes, Instruction Cycle, Interpretation and sequencing.	
4		**Control Unit Design**	6
	4.1	Hardwired Control Unit: State Table Method, Delay Element Methods.	
	4.2	Microprogrammed Control Unit: Micro Instruction-Format, Sequencing and execution, Micro operations, Examples of microprograms.	
5		**Memory Organization**	6
	5.1	Introduction and characteristics of memory, Types of RAM and ROM, Memory Hierarchy, 2-level Memory Characteristic,	
	5.2	Cache Memory: Concept, locality of reference, Design problems based on	

		mapping techniques, Cache coherence and write policies. Interleaved and Associative Memory.	
6		**Principles of Advanced Processor and Buses**	**8**
	6.1	Basic Pipelined Data path and control, data dependencies, data hazards, branch hazards, delayed branch, and branch prediction, Performance measures-CPI, Speedup, Efficiency, throughput, Amdhal's law.	
	6.2	Flynn's Classification, Introduction to multicore architecture.	
	6.3	Introduction to buses: ISA, PCI, USB. Bus Contention and Arbitration.	

Textbooks:	
1	R. P. Jain, "Modern Digital Electronic", McGraw-Hill Publication, 4thEdition.
2	William Stalling, "Computer Organization and Architecture: Designing and Performance", Pearson Publication 10TH Edition.
3	John P Hayes, "Computer Architecture and Organization", McGraw-Hill Publication, 3RD Edition.
4	Dr. M. Usha and T. S. Shrikanth, "Computer system Architecture and Organization", Wiley publication.

References:	
1	Andrew S. Tanenbaum, "Structured Computer Organization", Pearson Publication.
2	B. Govindarajalu, "Computer Architecture and Organization", McGraw-Hill Publication.
3	Malvino, "Digital computer Electronics", McGraw-Hill Publication, 3rdEdition.
4	Smruti Ranjan Sarangi, "Computer Organization and Architecture", McGraw-Hill 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.

Useful Links	
1	https://www.classcentral.com/course/swayam-computer-organization-and-architecture-a-pedagogical- aspect-9824
2	https://nptel.ac.in/courses/106/103/106103068/
3	https://www.coursera.org/learn/comparch
4	https://www.edx.org/learn/computer-architecture

Course Code	Course Name	Credits
CSC305	Computer Graphics	3

Prerequisite: Knowledge of C Programming and Basic Mathematics**.**		
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 and professional avenues in the field of Computer Graphics	
Course Outcomes: At the end of the course, the students should be able to		
1	Describe the basic concepts of Computer Graphics.	
2	Demonstrate various algorithms for basic graphics primitives.	
3	Apply 2-D geometric transformations on graphical objects.	
4	Use various Clipping algorithms on graphical objects	
5	Explore 3-D geometric transformations, curve representation techniques and projections methods.	
6	Explain visible surface detection techniques and Animation.	

Module		Detailed Content	Hours
1		**Introduction and Overview of Graphics System:**	**02**
	1.1	Definition and Representative uses of computer graphics, Overview of coordinate system, Definition of scan conversion, rasterization and rendering.	
	1.2	Raster scan & random scan displays, Architecture of raster graphics system with display processor, Architecture of random scan systems.	
2		**Output Primitives:**	**10**
	2.1	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)	
	2.2	Aliasing, Antialiasing techniques like Pre and post filtering, super sampling, and pixel phasing).	
	2.3	Filled Area Primitive: Scan line Polygon Fill algorithm, inside outside tests, Boundary Fill and Flood fill algorithm.	
3		**Two Dimensional Geometric Transformations**	**6**
	3.1	Basic transformations: Translation, Scaling, Rotation	
	3.2	Matrix representation and Homogeneous Coordinates	
	3.3	Composite transformation	
	3.4	Other transformations: Reflection and Shear	
4		**Two-Dimensional Viewing and Clipping**	**7**
	4.1	Viewing transformation pipeline and Window to Viewport coordinate transformation	
	4.2	Clipping operations: Point clipping, Line clipping algorithms: Cohen-Sutherland, Liang: Barsky, Polygon Clipping Algorithms: Sutherland-Hodgeman, Weiler-Atherton.	
5		**Three Dimensional Geometric Transformations, Curves and Fractal Generation**	**8**
	5.1	3D Transformations: Translation, Rotation, Scaling and Reflection	

	5.2	Composite transformations: Rotation about an arbitrary axis	
	5.3	Projections – Parallel, Perspective. (Matrix Representation)	
	5.4	Bezier Curve, B-Spline Curve, Fractal-Geometry: Fractal Dimension, Koch Curve.	
6		**Visible Surface Detection and Animation**	**6**
	6.1	Visible Surface Detection: Classification of Visible Surface Detection algorithm, Back Surface detection method, Depth Buffer method, Area Subdivision method	
	6.2	Animation: Introduction to Animation, Traditional Animation Techniques, Principles of Animation, Key framing: Character and Facial Animation, Deformation, Motion capture	

Textbooks:
1	Hearn & Baker, "Computer Graphics C version", 2nd Edition, Pearson Publication
2	James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice in C", 2nd Edition, Pearson Publication
3	Samit Bhattacharya, "Computer Graphics", Oxford Publication
References:
1	D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publications.
2	Zhigang Xiang, Roy Plastock, "Computer Graphics", Schaum"s Outlines McGraw-Hill Education
3	Rajesh K. Maurya, "Computer Graphics", Wiley India Publication.
4	F. S. Hill, "Computer Graphics using OpenGL", 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

Useful Links
1	https://www.classcentral.com/course/interactivegraphics-2067
2	https://swayam.gov.in/nd2_ntr20_ed15/preview
3	https://nptel.ac.in/courses/106/106/106106090/
4	https://www.edx.org/course/computer-graphics-2

Lab Code	Lab Name	Credit
CSL301	**Data Structures Lab**	**1**

Prerequisite: C Programming Language.

Lab Objectives:

1	To implement basic data structures such as arrays, linked lists, stacks and queues
2	Solve problem involving graphs, and trees
3	To develop application using data structure algorithms
4	Compute the complexity of various algorithms.

Lab Outcomes:

1	Students will be able to implement linear data structures & be able to handle operations like insertion, deletion, searching and traversing on them.
2	Students will be able to implement nonlinear data structures & be able to handle operations like insertion, deletion, searching and traversing on them
3	Students will be able to choose appropriate data structure and apply it in various problems
4	Students will be able to select appropriate searching techniques for given problems.

Suggested Experiments: Students are required to complete at least 10 experiments.

Star (*) marked experiments are compulsory.

Sr. No.	Name of the Experiment
1*	Implement Stack ADT using array.
2*	Convert an Infix expression to Postfix expression using stack ADT.
3*	Evaluate Postfix Expression using Stack ADT.
4	Applications of Stack ADT.
5*	Implement Linear Queue ADT using array.
6*	Implement Circular Queue ADT using array.
7	Implement Priority Queue ADT using array.
8*	Implement Singly Linked List ADT.
9*	Implement Circular Linked List ADT.
10	Implement Doubly Linked List ADT.
11*	Implement Stack / Linear Queue ADT using Linked List.
12*	Implement Binary Search Tree ADT using Linked List.
13*	Implement Graph Traversal techniques:) Depth First Search b) Breadth First Search
14	Applications of Binary Search Technique.

Useful Links:

1	www.leetcode.com
2	www.hackerrank.com
3	www.cs.usfca.edu/~galles/visualization/Algorithms.html
4	www.codechef.com

Term Work:

1	Term work should consist of 10 experiments.
2	Journal must include at least 2 assignments.
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance Theory & Practical: 05-marks, Assignments: 05-marks)

Oral & Practical exam

Based on the entire syllabus of CSL301and CSC303

Lab Code	Lab Name	Credit
CSL302	**Digital Logic & Computer Organization and Architecture Lab**	**1**

Prerequisite: C Programming Language.
Lab Objectives:

1	To implement operations of the arithmetic unit using algorithms.
2	Design and simulate different digital circuits.
3	To design memory subsystem including cache memory.
4	To demonstrate CPU and ALU design.
Lab Outcomes:	
1	To understand the basics of digital components
2	Design the basic building blocks of a computer: ALU, registers, CPU and memory
3	To recognize the importance of digital systems in computer architecture
4	To implement various algorithms for arithmetic operations.

List of Experiments:	
Sr. No.	**Name of the Experiment**
1	To verify the truth table of various logic gates using ICs.
2	To realize the gates using universal gates
3	Code conversion.
4	To realize half adder and full adder.
5	To implement logic operation using MUX IC.
6	To implement logic operation decoder IC.
7	Study of flip flop IC.
8	To implement ripple carry adder.
9	To implement carry look ahead adder.
10	To implement Booth's algorithm.
11	To implement restoring division algorithm.
12	To implement non restoring division algorithm.
13	To implement ALU design.
14	To implement CPU design.
15	To implement memory design.
16	To implement cache memory design.

Note:	
1	Any Four experiments from Exp. No. 1 to Exp. No. 7 using hardware.
2	Any Six experiments from Exp. No. 8 to Exp. No. 16 using Virtual Lab, expect Exp. No 10,11 and 12.
3	Exp. No. 10 to Exp. No. 12 using Programming language.
Digital Material:	
1	Manual to use Virtual Lab simulator for Computer Organization and Architecture developed by the Department of CSE, IIT Kharagpur.
2	Link http://cse10-iitkgp.virtual-labs.ac.in/

Term Work:	
1	Term work should consist of 10 experiments.
2	Journal must include at least 2 assignments on content of theory and practical of "Digital Logic &Computer Organization and Architecture"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.

| 4 | Total 25 Marks (Experiments: 15-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks) |

Course Code	Lab Name	Credits
CSL303	**Computer Graphics Lab**	**1**

Prerequisite: C Programming Language.

Lab Objectives:

1	Understand the need of developing graphics application
2	Learn algorithmic development of graphics primitives like line, circle, polygon etc.
3	Learn the representation and transformation of graphical images and pictures

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

1	Implement various output and filled area primitive algorithms
2	Apply transformation, projection and clipping algorithms on graphical objects.
3	Perform curve and fractal generation methods.
4	Develop a Graphical application/Animation 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++

List of Suggested Experiments:

Sr. No.	Name of the Experiment
1	Implement DDA Line Drawing algorithm (dotted/dashed/thick)
2	Implement Bresenham's Line algorithm(dotted/dashed/thick)
3	Implement midpoint Circle algorithm.
4	Implement midpoint Ellipse algorithm.
5	Implement Area Filling Algorithm: Boundary Fill, Flood Fill.
6	Implement Scan line Polygon Filling algorithm.
7	Implement Curve: Bezier for n control points, B Spline (Uniform)(at least one)
8	Implement Fractal generation method (anyone)
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 (at least one)
13	Program to perform 3D transformation.
14	Perform projection of a 3D object on Projection Plane: Parallel and Perspective.
15	Perform Animation (such as Rising Sun, Moving Vehicle, Smileys, Screen saver etc.)

Term Work:

1	Term work should consist of 10 experiments.
2	Journal must include at least 2 assignments
3	Mini Project to perform using C /C++/Java/OpenGL/Blender/ any other tool (2/3 students per group). Possible Ideas: Animation using multiple objects, Game development, Graphics editor: Like Paint brush, Text editor etc.
4	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
5	Total 25 Marks (Experiments: 10-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks, Mini Project: 5-marks)

Oral & Practical exam

	Based on the above contents and entire syllabus of CSC305

Lab Code	Lab Name	Credits
CSL304	**Skill based Lab Course: Object Oriented Programming with Java**	**2**

Prerequisite: Structured Programming Approach	

Lab Objectives:	
1	To learn the basic concepts of object-oriented programming
2	To study JAVA programming language
3	To study various concepts of JAVA programming like multithreading, exception Handling, packages, etc.
4	To explain components of GUI based programming.

Lab Outcomes: At the end of the course, the students should be able to	
1	To apply fundamental programming constructs.
2	To illustrate the concept of packages, classes and objects.
3	To elaborate the concept of strings, arrays and vectors.
4	To implement the concept of inheritance and interfaces.
5	To implement the concept of exception handling and multithreading.
6	To develop GUI based application.

Module		Detailed Content	Hours
1		**Introduction to Object Oriented Programming**	**2**
	1.1	OOP concepts: Objects, class, Encapsulation, Abstraction, Inheritance, Polymorphism, message passing.	
	1.2	Java Virtual Machine	
	1.3	Basic programming constructs: variables, data types, operators, unsigned right shift operator, expressions, branching and looping.	
2		**Class, Object, Packages and Input/output**	**6**
	2.1	Class, object, data members, member functions Constructors, types, static members and functions Method overloading Packages in java, types, user defined packages Input and output functions in Java, Buffered reader class, scanner class	
3		**Array, String and Vector**	**3**
	3.1	Array, Strings, String Buffer, Vectors	
4		**Inheritance**	**4**
	4.1	Types of inheritance, Method overriding, super, abstract class and abstract method, final, Multiple inheritance using interface, extends keyword	
5		**Exception handling and Multithreading**	**5**
	5.1	Exception handling using try, catch, finally, throw and throws, Multiple try and catch blocks, user defined exception Thread lifecycle, thread class methods, creating threads using extends and implements keyword.	
6		**GUI programming in JAVA**	**6**
	6.1	Applet and applet life cycle, creating applets, graphics class functions, parameter passing to applet, Font and color class. Event handling using event class AWT: working with windows, using AWT controls for GUI design Swing class in JAVA	

		Introduction to JDBC, JDBC-ODBC connectivity, JDBC architecture.	

Textbooks:

1	Herbert Schildt, 'JAVA: The Complete Reference', Ninth Edition, Oracle Press.
2	E. Balagurusamy, 'Programming with Java', McGraw Hill Education.

References:

1	Ivor Horton, "Beginning JAVA", Wiley India.
2	Dietal and Dietal, "Java: How to Program", 8th Edition, PHI .
3	"JAVA Programming", Black Book, Dreamtech Press.
4	"Learn to Master Java programming", Staredu solutions

Digital material:

1	www.nptelvideos.in
2	www.w3schools.com
3	www.tutorialspoint.com
4	https://starcertification.org/Certifications/Certificate/securejava

Suggested List of Programming Assignments/laboratory Work:

Sr. No.	Name of the Experiment
1	Programs on Basic programming constructs like branching and looping
2	Program on accepting input through keyboard.
3	Programs on class and objects
4	Program on method and constructor overloading.
5	Program on Packages
6	Program on 2D array, strings functions
7	Program on String Buffer and Vectors
8	Program on types of inheritance
9	Program on Multiple Inheritance
10	Program on abstract class and abstract methods.
11	Program using super and final keyword
12	Program on Exception handling
13	Program on user defined exception
14	Program on Multithreading
15	Program on Graphics class
16	Program on applet class
17	Program to create GUI application
18	Mini Project based on the content of the syllabus (Group of 2-3 students)

Term Work:

1	Term work should consist of 15 experiments.
2	Journal must include at least 2 assignments
3	Mini Project based on the content of the syllabus (Group of 2-3 students)
4	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
5	Total 50-Marks (Experiments: 15-marks, Attendance: 05-marks, Assignments: 05-marks, Mini Project: 20-marks, MCQ as a part of lab assignments: 5-marks)

Oral & Practical exam

Based on the entire syllabus of CSL 304: **Skill based Lab Course: Object Oriented Programming with Java**

Course code	Course Name	Credits
CSM301	**Mini Project A**	**02**

	Objectives
1	To acquaint with the process of identifying the needs and converting it into the problem.
2	To familiarize the process of solving the problem in a group.
3	To acquaint with the process of applying basic engineering fundamentals to attempt solutions to the problems.
4	To inculcate the process of self-learning and research.

	Outcome: Learner will be able to…
1	Identify problems based on societal /research needs.
2	Apply Knowledge and skill to solve societal problems in a group.
3	Develop interpersonal skills to work as member of a group or leader.
4	Draw the proper inferences from available results through theoretical/ experimental/simulations.
5	Analyze the impact of solutions in societal and environmental context for sustainable development.
6	Use standard norms of engineering practices
7	Excel in written and oral communication.
8	Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
9	Demonstrate project management principles during project work.

	Guidelines for Mini Project
1	Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.
2	Students should do survey and identify needs, which shall be converted into problem statement for mini project in consultation with faculty supervisor/head of department/internal committee of faculties.
3	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of mini project.
4	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
5	Faculty supervisor may give inputs to students during mini project activity; however, focus shall be on self-learning.
6	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
7	Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
8	The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.
9	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the Mini Projects, it is preferable that a single project of appropriate level and quality to be carried out in two semesters by all the groups of the students. i.e. Mini Project 1 in semester III and IV. Similarly, Mini Project 2 in semesters V and VI.
10	However, based on the individual students or group capability, with the mentor's recommendations, if the proposed Mini Project adhering to the qualitative aspects mentioned above gets completed in odd semester, then that group can be allowed to work on the extension of the Mini Project with suitable improvements/modifications or a completely new project idea in even semester. This policy can be adopted on case by case basis.

Term Work	
The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of mini project to be evaluated on continuous basis, minimum two reviews in each semester.	
In continuous assessment focus shall also be on each individual student, assessment based on individual's contribution in group activity, their understanding and response to questions.	
Distribution of Term work marks for both semesters shall be as below:	**Marks**
1	Marks awarded by guide/supervisor based on logbook	10
2	Marks awarded by review committee	10
3	Quality of Project report	05

Review / progress monitoring committee may consider following points for assessment based on either one year or half year project as mentioned in general guidelines	

One-year project:

1	In first semester entire theoretical solution shall be ready, including components/system selection and cost analysis. Two reviews will be conducted based on presentation given by students group. • First shall be for finalization of problem • Second shall be on finalization of proposed solution of problem.
2	In second semester expected work shall be procurement of component's/systems, building of working prototype, testing and validation of results based on work completed in an earlier semester. • First review is based on readiness of building working prototype to be conducted. • Second review shall be based on poster presentation cum demonstration of working model in last month of the said semester.

Half-year project:

1	In this case in one semester students' group shall complete project in all aspects including, • Identification of need/problem • Proposed final solution • Procurement of components/systems • Building prototype and testing
2	Two reviews will be conducted for continuous assessment, • First shall be for finalization of problem and proposed solution • Second shall be for implementation and testing of solution.

Assessment criteria of Mini Project.

Mini Project shall be assessed based on following criteria;

1	Quality of survey/ need identification
2	Clarity of Problem definition based on need.
3	Innovativeness in solutions
4	Feasibility of proposed problem solutions and selection of best solution
5	Cost effectiveness
6	Societal impact
7	Innovativeness
8	Cost effectiveness and Societal impact
9	Full functioning of working model as per stated requirements

10	Effective use of skill sets
11	Effective use of standard engineering norms
12	Contribution of an individual's as member or leader
13	Clarity in written and oral communication
	In **one year, project**, first semester evaluation may be based on first six criteria's and remaining may be used for second semester evaluation of performance of students in mini project.
	In case of **half year project** all criteria's in generic may be considered for evaluation of performance of students in mini project.

Guidelines for Assessment of Mini Project Practical/Oral Examination:

1	Report should be prepared as per the guidelines issued by the University of Mumbai.
2	Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organizations having experience of more than five years approved by head of Institution.
3	Students shall be motivated to publish a paper based on the work in Conferences/students competitions.

Mini Project shall be assessed based on following points;

1	Quality of problem and Clarity
2	Innovativeness in solutions
3	Cost effectiveness and Societal impact
4	Full functioning of working model as per stated requirements
5	Effective use of skill sets
6	Effective use of standard engineering norms
7	Contribution of an individual's as member or leader
8	Clarity in written and oral communication

Course Code	Course Name	Credits
CSC401	Engineering Mathematics-IV	4

Pre-requisite: Engineering Mathematics-I, Engineering Mathematics-II, Engineering Mathematics-III, Binomial Distribution.

Course Objectives: The course aims to learn:

1	Matrix algebra to understand engineering problems.
2	Line and Contour integrals and expansion of a complex valued function in a power series.
3	Z-Transforms and Inverse Z-Transforms with its properties.
4	The concepts of probability distributions and sampling theory for small samples.
5	Linear and Non-linear programming problems of optimization.

Course Outcomes: On successful completion, of course, learner/student will be able to:

1	Apply the concepts of eigenvalues and eigenvectors in engineering problems.
2	Use the concepts of Complex Integration for evaluating integrals, computing residues & evaluate various contour integrals.
3	Apply the concept of Z- transformation and inverse in engineering problems.
4	Use the concept of probability distribution and sampling theory to engineering problems.
5	Apply the concept of Linear Programming Problems to optimization.
6	Solve Non-Linear Programming Problems for optimization of engineering problems.

Module	Detailed Contents		Hours
1	**Linear Algebra (Theory of Matrices)**		7
	1.1	Characteristic Equation, Eigenvalues and Eigenvectors, and properties (without proof)	
	1.2	Cayley-Hamilton Theorem (without proof), verification and reduction of higher degree polynomials	
	1.3	Similarity of matrices, diagonalizable and non-diagonalizable matrices	
	1.4	**Self-learning Topics:** Derogatory and non-derogatory matrices, Functions of Square Matrix, Linear Transformations, Quadratic forms.	
2	**Complex Integration**		7
	2.1	Line Integral, Cauchy's Integral theorem for simple connected and multiply connected regions (without proof), Cauchy's Integral formula (without proof).	
	2.2	Taylor's and Laurent's series (without proof).	
	2.3	Definition of Singularity, Zeroes, poles of $f(z)$, Residues, Cauchy's Residue Theorem (without proof)	
	2.4	**Self-learning Topics:** Application of Residue Theorem to evaluate real integrations.	
3	**Z Transform**		5
	3.1	Definition and Region of Convergence, Transform of Standard Functions: $\{\Box^\Box\Box^\Box\}, \{\Box^{|\Box|}\}, \{\Box^{\Box+\Box}\Box.\Box^\Box\}, \{\Box^\Box \sin(\Box\Box+\Box)\}, \{\Box^\Box \sinh \Box\Box\}, \{\Box^\Box \cosh \Box\Box\}$.	
	3.2	Properties of Z Transform: Change of Scale, Shifting Property, Multiplication, and Division by k, Convolution theorem.	
	3.3	Inverse Z transform: Partial Fraction Method, Convolution Method.	
	3.4	**Self-learning Topics:** Initial value theorem, Final value theorem, Inverse of Z Transform by Binomial Expansion	
4	**Probability Distribution and Sampling Theory**		7
	4.1	Probability Distribution: Poisson and Normal distribution	

	4.2	Sampling distribution, Test of Hypothesis, Level of Significance, Critical region, One-tailed, and two-tailed test, Degree of freedom.	
	4.3	Students' t-distribution (Small sample). Test the significance of mean and Difference between the means of two samples. Chi-Square Test: Test of goodness of fit and independence of attributes, Contingency table.	
	4.4	**Self-learning Topics:** Test significance for Large samples, Estimate parameters of a population, Yate's Correction.	
5	**Linear Programming Problems**		6
	5.1	Types of solutions, Standard and Canonical of LPP, Basic and Feasible solutions, slack variables, surplus variables, Simplex method.	
	5.2	Artificial variables, Big-M method (Method of penalty)	
	5.3	Duality, Dual of LPP and Dual Simplex Method	
	5.4	**Self-learning Topics:** Sensitivity Analysis, Two-Phase Simplex Method, Revised Simplex Method.	
6	**Nonlinear Programming Problems**		7
	6.1	NLPP with one equality constraint (two or three variables) using the method of Lagrange's multipliers	
	6.2	NLPP with two equality constraints	
	6.3	NLPP with inequality constraint: Kuhn-Tucker conditions	
	6.4	**Self-learning Topics:** Problems with two inequality constraints, Unconstrained optimization: One-dimensional search method (Golden Search method, Newton's method). Gradient Search method	

References:	
1	Erwin Kreyszig, "Advanced Engineering Mathematics", John Wiley & Sons.
2	R. K. Jain and S. R. K. Iyengar, "Advanced Engineering Mathematics", Narosa.
3	Brown and Churchill, "Complex Variables and Applications", McGraw-Hill Education.
4	T. Veerarajan, "Probability, Statistics and Random Processes", McGraw-Hill Education.
5	Hamdy A Taha, "Operations Research: An Introduction", Pearson.
6	S.S. Rao, "Engineering Optimization: Theory and Practice", Wiley-Blackwell.
7	Hira and Gupta, "Operations Research", S. Chand Publication.

Term Work:		
General Instructions:		
1	Batch wise tutorial shave to be conducted. The number of students per batch will be as per University pattern for practical.	
2	Students must be encouraged to write at least 6 class tutorials on the entire syllabus.	
3	A group of 4-6 students should be assigned a self-learning topic. Students should prepare a presentation/problem solving of 10-15 minutes. This will be considered as a mini project in Engineering Mathematics. This project will be graded out of 10 marks depending on the performance of the students.	
The distribution of Term Work marks will be as follows:		
1	Attendance (Theory and Tutorial)	05 marks
2	Class Tutorials on entire syllabus	10 marks
3	Mini project	10 marks

Assessment:
Internal Assessment Test:
The assessment consists of two class tests of 20 marks each. The 1stclass test (Internal Assessment I) has to be conducted when approximately 40% of the syllabus is completed. The 2nd class test has to be conducted (Internal Assessment II) when an additional 35% syllabus is

completed. The duration of each test will be for one hour.	
End Semester Theory Examination:	
1	The question paper will comprise a total of 6 questions, each carrying 20 marks.
2	Out of the 6 questions, 4 questions have to be attempted.
3	Question 1, based on the entire syllabus, will have 4sub-questions of 5 marks each and is compulsory.
4	Question 2 to Question 6 will have 3 sub-questions, each of 6, 6, and 8 marks, respectively.
5	Each sub-question in (4) will be from different modules of the syllabus.
6	Weightage of each module will be proportional to the number of lecture hours, as mentioned in the syllabus.

Course Code	Course Name	Credit
CSC402	**Analysis of Algorithms**	**3**

Prerequisite: Data structure concepts, Discrete structures	
Course Objectives:	
1	To provide mathematical approaches for Analysis of Algorithms
2	To understand and solve problems using various algorithmic approaches
3	To analyze algorithms using various methods
Course Outcomes: At the end of the course learner 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.
6	Explain and apply string matching techniques.

Module		Detailed Contents	Hours
1		**Introduction**	8
	1.1	Performance analysis, space, and time complexity Growth of function, Big-Oh, Omega Theta notation Mathematical background for algorithm analysis. Complexity class: Definition of P, NP, NP-Hard, NP-Complete Analysis of selection sort, insertion sort.	
	1.2	Recurrences: The substitution method, Recursion tree method, Master method	
2		**Divide and Conquer Approach**	6
	2.1	General method, Merge sort, Quick sort, Finding minimum and maximum algorithms and their Analysis, Analysis of Binary search.	
3		**Greedy Method Approach**	6
	3.1	General Method, Single source shortest path: Dijkstra Algorithm Fractional Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees: Kruskal and Prim's algorithms	
4		**Dynamic Programming Approach**	9
	4.1	General Method, Multistage graphs, Single source shortest path: Bellman Ford Algorithm All pair shortest path: Floyd Warshall Algorithm, Assembly-line scheduling Problem0/1 knapsack Problem, Travelling Salesperson problem, Longest common subsequence	
5		**Backtracking and Branch and bound**	6
	5.1	General Method, Backtracking: N-queen problem, Sum of subsets, Graph coloring	
	5.2	Branch and Bound: Travelling Salesperson Problem, 15 Puzzle problem	
6		**String Matching Algorithms**	4
	6.1	The Naïve string-matching algorithm, The Rabin Karp algorithm, The Knuth-Morris-Pratt algorithm	

Textbooks:	
1	T. H. Cormen, 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.

References:

1	Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw-Hill Edition.
2	S. K. Basu, "Design Methods and Analysis of Algorithm", PHI

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 total six questions.
2	All question carries equal marks
3	Questions will be mixed in nature (for example supposed Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)
4	Only Four question need to be solved.
5	In question paper weightage of each module will be proportional to number of respective lecture hours as mention in the syllabus.

Useful Links

1	https://nptel.ac.in/courses/106/106/106106131/
2	https://swayam.gov.in/nd1_noc19_cs47/preview
3	https://www.coursera.org/specializations/algorithms
4	https://www.mooc-list.com/tags/algorithms

Course Code:	Course Title	Credit
CSC403	Database Management System	3

Prerequisite: Data Structures	
Course Objectives:	
1	Develop entity relationship data model and its mapping to relational model
2	Learn relational algebra and Formulate SQL queries
3	Apply normalization techniques to normalize the database
4	Understand concept of transaction, concurrency control and recovery techniques.
Course Outcomes:	
1	Recognize the need of database management system
2	Design ER and EER diagram for real life applications
3	Construct relational model and write relational algebra queries.
4	Formulate SQL queries
5	Apply the concept of normalization to relational database design.
6	Describe the concept of transaction, concurrency and recovery.

Module		Content	Hrs
1		**Introduction Database Concepts**	**3**
	1.1	Introduction, Characteristics of databases, File system v/s Database system, Data abstraction and data Independence, DBMS system architecture, Database Administrator	
2		**Entity–Relationship Data Model**	**6**
	2.1	The Entity-Relationship (ER) Model: Entity types: Weak and strong entity sets, Entity sets, Types of Attributes, Keys, Relationship constraints: Cardinality and Participation, Extended Entity-Relationship (EER) Model: Generalization, Specialization and Aggregation	
3		**Relational Model and relational Algebra**	**8**
	3.1	Introduction to the Relational Model, relational schema and concept of keys. Mapping the ER and EER Model to the Relational Model, Relational Algebra-operators, Relational Algebra Queries.	
4		**Structured Query Language (SQL)**	**6**
	4.1	Overview of SQL, Data Definition Commands, Integrity constraints: key constraints, Domain Constraints, Referential integrity , check constraints, Data Manipulation commands, Data Control commands, Set and string operations, aggregate function-group by, having, Views in SQL, joins, Nested and complex queries, Triggers	
5		**Relational-Database Design**	**6**
	5.1	Pitfalls in Relational-Database designs, Concept of normalization, Function Dependencies, First Normal Form, 2NF, 3NF, BCNF.	
6		**Transactions Management and Concurrency and Recovery**	**10**
	6.1	Transaction concept, Transaction states, ACID properties, Transaction Control Commands, Concurrent Executions, Serializability-Conflict and View, Concurrency Control: Lock-based, Timestamp-based protocols, Recovery System: Log based recovery, Deadlock handling	

Textbooks:	
1	Korth, Slberchatz, Sudarshan, Database System Concepts, 6thEdition, McGraw Hill
2	Elmasri and Navathe, Fundamentals of Database Systems, 5thEdition, Pearson Education
3	Raghu Ramkrishnan and Johannes Gehrke, Database Management Systems, TMH
References:	
1	Peter Rob and Carlos Coronel, Database Systems Design, Implementation and Management‖, Thomson Learning, 5thEdition.
2	Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g, Black Book, Dreamtech Press.
3	G. K. Gupta, Database Management Systems, McGraw Hill, 2012

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 total six questions.
2	All question carries equal marks
3	Questions will be mixed in nature (for example supposed Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)
4	Only Four question need to be solved.
5	In question paper weightage of each module will be proportional to number of respective lecture hours as mention in the syllabus.

Useful Links	
1	https://nptel.ac.in/courses/106/105/106105175/
2	https://swayam.gov.in/nd1_noc19_cs46/preview
3	https://www.classcentral.com/course/swayam-database-management-system-9914
4	https://www.mooc-list.com/tags/dbms

Course Code	Course Name	Credit
CSC404	**Operating System**	**03**

Prerequisites: Data structures and Computer architecture

Course Objectives:

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

Course Outcome:

1	Understand the objectives, functions and structure of OS
2	Analyze the concept of process management and evaluate performance of processscheduling algorithms.
3	Understand and apply the concepts of synchronization and deadlocks
4	Evaluate performance of Memory allocation and replacement policies
5	Understand the concepts of file management.
	Apply concepts of I/O management and analyze techniques of disk scheduling.

Module	Detailed Content		Hours
1	**Operating system Overview**		**4**
	1.1	Introduction, Objectives, Functions and Evolution of Operating System	
	1.2	Operating system structures: Layered, Monolithic and Microkernel	
	1.3	Linux Kernel, Shell and System Calls	
2	**Process and Process Scheduling**		**9**
	2.1	Concept of a Process, Process States, Process Description, Process Control Block.	
	2.2	Uniprocessor Scheduling-Types: Preemptive and Non-preemptive scheduling algorithms (FCFS, SJF, SRTN, Priority, RR)	
	2.3	Threads: Definition and Types, Concept of Multithreading	
3	**Process Synchronization and Deadlocks**		**9**
	3.1	Concurrency: Principles of Concurrency, Inter-Process Communication, Process Synchronization.	
	3.2	Mutual Exclusion: Requirements, Hardware Support (TSL), Operating System Support (Semaphores), Producer and Consumer problem.	
	3.3	Principles of Deadlock: Conditions and Resource, Allocation Graphs, Deadlock Prevention, Deadlock Avoidance: Banker‟s Algorithm, Deadlock Detection and Recovery, Dining Philosophers Problem.	
4	**Memory Management**		**9**
	4.1	Memory Management Requirements, Memory Partitioning: Fixed, Partitioning, Dynamic Partitioning, Memory Allocation Strategies: Best-Fit, First Fit, Worst Fit, Paging and Segmentation, TLB	
	4.2	Virtual Memory: Demand Paging, Page Replacement Strategies: FIFO, Optimal, LRU, Thrashing	
5		**File Management**	**4**

	5.1	Overview, File Organization and Access, File Directories, File Sharing	
6		**I/O management**	**4**
	6.1	I/O devices, Organization of the I/O Function, Disk Organization, I/O Management and Disk Scheduling: FCFS, SSTF, SCAN, CSCAN, LOOK, C-LOOK.	

Textbooks:	
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
References:	
1	Achyut Godbole and Atul Kahate, Operating Systems, McGraw Hill Education, 3rd Edition
2	Andrew Tannenbaum, Operating System Design and Implementation, Pearson, 3rd Edition.
3	Maurice J. Bach, "Design of UNIX Operating System", PHI
4	Sumitabha Das, "UNIX: Concepts and Applications", McGraw Hill, 4th Edition

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

Useful Links	
1	https://swayam.gov.in/nd1_noc19_cs50/preview
2	https://nptel.ac.in/courses/117/106/117106113/
3	https://www.classcentral.com/course/swayam-introduction-to-operating-systems-6559

Course Code	Course Name	Credits
CSC405	**Microprocessor**	**3**

Prerequisites: Digital Logic and Computer Architecture	

Course objectives:

1	To equip students with the fundamental knowledge and basic technical competence in thefield of Microprocessors.
2	To emphasize on instruction set and logic to build assembly language programs.
3	To prepare students for higher processor architectures and embedded systems

Course outcomes: On successful completion of course, learner will be able to:

1	Describe core concepts of 8086 microprocessor.
2	Interpret the instructions of 8086 and write assembly and Mixed language programs.
3	Identify the specifications of peripheral chip.
4	Design 8086 based system using memory and peripheral chips.
5	Appraise the architecture of advanced processors
6	Understand hyperthreading technology

Module	Detailed Contents		Hours
1	**The Intel Microprocessors 8086 Architecture**		8
	1.1	8086CPU Architecture,	
	1.2	Programmer's Model	
	1.3	Functional Pin Diagram	
	1.4	Memory Segmentation	
	1.5	Banking in 8086	
	1.6	Demultiplexing of Address/Data bus	
	1.7	Functioning of 8086 in Minimum mode and Maximum mode	
	1.8	Timing diagrams for Read and Write operations in minimum and maximum mode	
	1.9	Interrupt structure and its servicing	
2	**Instruction Set and Programming**		**6**
	2.1	Addressing Modes	
	2.2	Instruction set-Data Transfer Instructions, String Instructions, Logical Instructions, Arithmetic Instructions, Transfer of Control Instructions, Processor Control Instructions	
	2.3	Assembler Directives and Assembly Language Programming, Macros, Procedures	
3	**Memory and Peripherals interfacing**		**8**
	3.1	Memory Interfacing - RAM and ROM Decoding Techniques – Partial and Absolute	
	3.2	8255-PPI-Block diagram, CWR, operating modes, interfacing with 8086.	
	3.3	8257-DMAC-Block diagram, DMA operations and transfer modes.	
	3.4	Programmable Interrupt Controller 8259-Block Diagram, Interfacing the 8259 in single and cascaded mode.	
4	**Intel 80386DX Processor**		**7**
	4.1	Architecture of 80386 microprocessor	
	4.2	80386 registers–General purpose Registers, EFLAGS and Control	

		registers	
	4.3	Real mode, Protected mode, virtual 8086 mode	
	4.4	80386 memory management in Protected Mode – Descriptors and selectors, descriptor tables, the memory paging mechanism	
5		**Pentium Processor**	**6**
	5.1	Pentium Architecture	
	5.2	Superscalar Operation,	
	5.3	Integer &Floating-Point Pipeline Stages,	
	5.4	Branch Prediction Logic,	
	5.5	Cache Organization and	
	5.6	MESI protocol	
6		**Pentium 4**	**4**
	6.1	Comparative study of 8086, 80386, Pentium I, Pentium II and Pentium III	
	6.2	Pentium 4: Net burst micro architecture.	
	6.3	Instruction translation look aside buffer and branch prediction	
	6.4	Hyper threading technology and its use in Pentium 4	

Textbooks:	
1	John Uffenbeck, "8086/8088 family: Design Programming and Interfacing", PHI.
2	Yu-Cheng Liu, Glenn A. Gibson, "Microcomputer System: The 8086/8088 Family, Architecture, Programming and Design", Prentice Hall
3	Walter A. Triebel, "The 80386DX Microprocessor: hardware, Software and Interfacing", Prentice Hall
4	Tom Shanley and Don Anderson, "Pentium Processor System Architecture", Addison-Wesley.
5	K. M. Bhurchandani and A. K. Ray, "Advanced Microprocessors and Peripherals", McGraw Hill

References:	
1	Barry B. Brey, "Intel Microprocessors", 8[th]Edition, Pearson Education India
2	Douglas Hall, "Microprocessor and Interfacing", Tata McGraw Hill.
3	Intel Manual
4	Peter Abel, "IBM PC Assembly language and Programming", 5th Edition, PHI
5	James Antonakons, "The Pentium Microprocessor", Pearson Education

Assessment:

Internal Assessment Test:
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.

Useful Links	
1	https://swayam.gov.in/nd1_noc20_ee11/preview
2	https://nptel.ac.in/courses/108/105/108105102/
3	https://www.classcentral.com/course/swayam-microprocessors-and-microcontrollers-9894
4	https://www.mooc-list.com/tags/microprocessors

Course Name	Lab Name	Credit
CSL401	**Analysis of Algorithms Lab**	**1**

Prerequisite: Basic knowledge of programming and data structure

Lab Objectives:	
1	To introduce the methods of designing and analyzing algorithms
2	Design and implement efficient algorithms for a specified application
3	Strengthen the ability to identify and apply the suitable algorithm for the given real-world problem.
4	Analyze worst-case running time of algorithms and understand fundamental algorithmic problems.

Lab Outcomes: At the end of the course, the students will be able to	
1	Implement the algorithms using different approaches.
2	Analyze the complexities of various algorithms.
3	Compare the complexity of the algorithms for specific problem.

Description		
Implementation can be in any language.		
Suggested Practical List:		
Sr No		**Suggested Experiment List**
1		**Introduction**
	1.1	Selection sort, Insertion sort
2		**Divide and Conquer Approach**
	2.1	Finding Minimum and Maximum, Merge sort, Quick sort, Binary search
3		**Greedy Method Approach**
	3.1	Single source shortest path- Dijkstra
		Fractional Knapsack problem
		Job sequencing with deadlines
		Minimum cost spanning trees-Kruskal and Prim's algorithm
4		**Dynamic Programming Approach**
	4.1	Single source shortest path- Bellman Ford
		All pair shortest path- Floyd Warshall
		0/1 knapsack
		Travelling salesperson problem
		Longest common subsequence
5		**Backtracking and Branch and bound**
	5.1	N-queen problem
		Sum of subsets
		Graph coloring
6		**String Matching Algorithms**
	6.1	The Naïve string-matching Algorithms
		The Rabin Karp algorithm
		The Knuth-Morris-Pratt algorithm

Term Work:	
1	Term work should consist of 10 experiments.
2	Journal must include at least 2 assignments on content of theory and practical of "Analysis of Algorithms"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks)
Oral & Practical exam	
	Based on the entire syllabus of CSC402: Analysis of Algorithms

Lab Code	Lab Name	Credit
CSL402	**Database Management System Lab**	**1**

Prerequisite: Discrete Structures

Lab Objectives:

1	To explore design and develop of relational model
2	To present SQL and procedural interfaces to SQL comprehensively
3	To introduce the concepts of transactions and transaction processing

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

1	Design ER /EER diagram and convert to relational model for the realworld application.
2	Apply DDL, DML, DCL and TCL commands
3	Write simple and complex queries
4	UsePL / SQL Constructs.
5	Demonstrate the concept of concurrent transactions execution and frontend-backend connectivity

Suggested List of Experiments	
Sr. No.	**Title of Experiment**
1	Identify the case study and detail statement of problem. Design an Entity-Relationship (ER) / Extended Entity-Relationship (EER) Model.
2	Mapping ER/EER to Relational schema model.
3	Create a database using Data Definition Language (DDL) and apply integrity constraints for the specified System
4	Apply DML Commands for the specified system
5	Perform Simple queries, string manipulation operations and aggregate functions.
6	Implement various Join operations.
7	Perform Nested and Complex queries
8	Perform DCL and TCL commands
9	Implement procedure and functions
10	Implementation of Views and Triggers.
11	Demonstrate Database connectivity
12	Implementation and demonstration of Transaction and Concurrency control techniques using locks.

Term Work:	
1	Term work should consist of 10 experiments.
2	Journal must include at least 2 assignments on content of theory and practical of "Database Management System"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks)

Oral & Practical exam : Based on the entire syllabus of CSC403: Database Management System

Course Code	Course Name	Credit
CSL403	**Operating System Lab**	**01**

Prerequisite: Knowledge on Operating system principles

Lab Objectives:

1	To gain practical experience with designing and implementing concepts of operating systems such as system calls, CPU scheduling, process management, memory management, file systems and deadlock handling using C language in Linux environment.
2	To familiarize students with the architecture of Linux OS.
3	To provide necessary skills for developing and debugging programs in Linux environment.
4	To learn programmatically to implement simple operation system mechanisms

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

1	Demonstrate basic Operating system Commands, Shell scripts, System Calls and API wrt Linux
2	Implement various process scheduling algorithms and evaluate their performance.
3	Implement and analyze concepts of synchronization and deadlocks.
4	Implement various Memory Management techniques and evaluate their performance.
5	Implement and analyze concepts of virtual memory.
6	Demonstrate and analyze concepts of file management and I/O management techniques.

Suggested List of Experiments		
Sr. No.		**Content**
1		**Explore Linux Commands**
	1.1	**Explore usage of basic Linux Commands and system calls for file, directory and process management.** For eg: (mkdir, chdir, cat, ls, chown, chmod, chgrp, ps etc. system calls: open, read, write, close, getpid, setpid, getuid, getgid, getegid, geteuid. sort, grep, awk, etc.)
2		**Linux shell script**
	2.1	Write shell scripts to do the following: a. Display OS version, release number, kernel version b. Display top 10 processes in descending order c. Display processes with highest memory usage. d. Display current logged in user and log name. Display current shell, home directory, operating system type, current path setting, current working directory.
3		**Linux- API**
	3.1	Implement any one basic commands of linux like ls, cp, mv and others using kernel APIs.
4		**Linux- Process**
	4.1	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. b. Explore wait and waitpid before termination of process.
5		**Process Management: Scheduling**

	5.1	a. Write a program to demonstrate the concept of non-preemptive scheduling algorithms.
		b. Write a program to demonstrate the concept of preemptive scheduling algorithms
6		**Process Management: Synchronization**
	6.1	**a.** Write a C program to implement solution of Producer consumer problem through Semaphore
7		**Process Management: Deadlock**
	7.1	a. Write a program to demonstrate the concept of deadlock avoidance through Banker's Algorithm
		b. Write a program demonstrate the concept of Dining Philospher's Problem
8		**Memory Management**
	8.1	a. Write a program to demonstrate the concept of MVT and MFT memory management techniques
		b. Write a program to demonstrate the concept of dynamic partitioning placement algorithms i.e. Best Fit, First Fit, Worst-Fit etc.
9		**Memory Management: Virtual Memory**
	9.1	a. Write a program to demonstrate the concept of demand paging for simulation of Virtual Memory implementation
		b. Write a program in C demonstrate the concept of page replacement policies for handling page faults eg: FIFO, LRU etc.
10		**File Management & I/O Management**
	10.1	a. Write a C program to simulate File allocation strategies typically sequential, indexed and linked files
		b. Write a C program to simulate file organization of multi-level directory structure.
		c. Write a program in C to do disk scheduling - FCFS, SCAN, C-SCAN

Term Work:	
1	Term work should consist of 10 experiments covering all modules.
2	Journal must include at least 2 assignments on content of theory and practical of "Database Management System"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks)
Oral & Practical exam	
	Based on the entire syllabus of CSC405: Operating System.

Lab Code	Lab Name	Credits
CSL404	**Microprocessor Lab**	**1**

Prerequisite: Basic knowledge digital integrated circuits

Lab Objectives:	
1	To emphasize on use of Assembly language program.
2	To prepare students for advanced subjects like embedded system and IOT.

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

1	Use appropriate instructions to program microprocessor to perform various task
2	Develop the program in assembly/ mixed language for Intel 8086 processor
3	Demonstrate the execution and debugging of assembly/ mixed language program

Suggested List of Experiments:	
Sr. No.	**Title of Experiments**
1	Use of programming tools (Debug/TASM/MASM/8086kit) to perform basic arithmetic operations on 8-bit/16-bit data
2	Code conversion (Hex to BCD and BCD to Hex)/ (ASCII to BCD and BCD to ASCII)
3	Assembly programming for 16-bit addition, subtraction, multiplication and division (menu based)
4	Assembly program based on string instructions (overlapping/non-overlapping block transfer/ string search/ string length)
5	Assembly program to display the contents of the flag register.
6	Any Mixed Language programs.
7	Assembly program to find the GCD/ LCM of two numbers
8	Assembly program to sort numbers in ascending/ descending order
9	Any program using INT 10H
10	Assembly program to find minimum/ maximum number from a given array.
11	Assembly Program to display a message in different color with blinking
12	Assembly program using procedure.
13	Assembly program using macro.
14	Program and interfacing using 8255.
15	Program and interfacing of ADC/ DAC/ Stepper motor.

Term Work:	
1	Term work should consist of 10 experiments, out of theses at least one experiment on hardware interfacing.
2	Journal must include at least 2 assignments on content of theory and practical of "Microprocessor"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks)

Lab Code	Lab Name	Credit
CSL405	**Skill Base Lab Course: Python Programming**	**2**

Prerequisite: Knowledge of some programming language like C, Java	

Lab Objectives:	
1	Basics of Python programming
2	Decision Making, Data structure and Functions in Python
3	Object Oriented Programming using Python
4	Web framework for developing

Lab Outcomes: At the end of the course, the students will be able to	
1	To understand basic concepts in python.
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 Multithreading concepts using python.

Module		Detailed Content	Hours
1		**Python basics**	**5**
	1.1	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**	**4**
	2.1	Files in Python, Directories, Building Modules, Packages, Text Processing, Regular expression in python.	
3		**Data Structure in Python**	**3**
	3.1	Link List, Stack, Queues, Dequeues	
4		**Python Integration Primer**	**4**
	4.1	Graphical User interface, Networking in Python, Python database connectivity, Introduction to Django	
5		**Multithreading**	**4**
	5.1	Thread and Process, Starting a thread, Threading module, Synchronizing threads, Multithreaded Priority Queue	
6		**NumPy and Pandas**	**6**
	6.1	Creating NumPy arrays, Indexing and slicing in NumPy, creating multidimensional arrays, NumPy Data types, Array Attribute, Indexing and Slicing, Creating array views copies, Manipulating array shapes I/O	
	6.2	Basics of Pandas, Using multilevel series, Series and Data Frames, Grouping, aggregating, Merge Data Frames	

Textbooks:	
1	Dr. R. Nageswara Rao, "Core Python Programming", Dreamtech Press
2	Beginning Python: Using Python 2.6 and Python 3.1. James Payne, Wrox Publication
3	Anurag Gupta, G. P. Biswas, "Python Programming", McGraw-Hill
4	E. Balagurusamy, "Introduction to computing and problem-solving using python", McGraw Hill Education

References:	
1	Learn Python the Hard Way, 3rd Edition, Zed Shaw's Hard Way Series

2	Laura Cassell, Alan Gauld, "Python Projects", 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	https://starcertification.org/Certifications/Certificate/python

Suggested experiments using Python:	
Sr. No.	**Title of Experiments**
1	Exploring basics of python like data types (strings, list, array, dictionaries, set, tuples) and control statements.
2	Creating functions, classes and objects using python. Demonstrate exception handling and inheritance.
3	Exploring Files and directories a. Python program to append data to existing file and then display the entire file b. Python program to count number of lines, words and characters in a file. c. Python program to display file available in current directory
4	Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes.
5	Menu driven program for data structure using built in function for link list, stack and queue.
6	Program to demonstrate CRUD (**create, read, update and delete)** operations on database (SQLite/ MySQL) using python.
7	Creation of simple socket for basic information exchange between server and client.
8	Creating web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression).
9	Programs on Threading using python.
10	Exploring basics of NumPy Methods.
11	Program to demonstrate use of NumPy: Array objects.
12	Program to demonstrate Data Series and Data Frames using Pandas.
13	Program to send email and read content of URL.

Term Work:	
1	Term work should consist of 12 experiments.
2	Journal must include at least 2 assignments
3	Mini Project based on the content of the syllabus (Group of 2-3 students)
4	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
5	Total 25 Marks (Journal: 10-marks, Attendance: 05-marks, and Mini Project: 10-marks)

Course code	Course Name	Credits
CSM401	**Mini Project B**	**02**

Objectives

1	To acquaint with the process of identifying the needs and converting it into the problem.
2	To familiarize the process of solving the problem in a group.
3	To acquaint with the process of applying basic engineering fundamentals to attempt solutions to the problems.
4	To inculcate the process of self-learning and research.

Outcome: Learner will be able to…

1	Identify problems based on societal /research needs.
2	Apply Knowledge and skill to solve societal problems in a group.
3	Develop interpersonal skills to work as member of a group or leader.
4	Draw the proper inferences from available results through theoretical/ experimental/simulations.
5	Analyze the impact of solutions in societal and environmental context for sustainable development.
6	Use standard norms of engineering practices
7	Excel in written and oral communication.
8	Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
9	Demonstrate project management principles during project work.

Guidelines for Mini Project

1	Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.
2	Students should do survey and identify needs, which shall be converted into problem statement for mini project in consultation with faculty supervisor/head of department/internal committee of faculties.
3	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of mini project.
4	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
5	Faculty supervisor may give inputs to students during mini project activity; however, focus shall be on self-learning.
6	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
7	Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
8	The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.
9	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the Mini Projects, it is preferable that a single project of appropriate level and quality to be carried out in two semesters by all the groups of the students. i.e. Mini Project 1 in semester III and IV. Similarly, Mini Project 2 in semesters V and VI.
10	However, based on the individual students or group capability, with the mentor's recommendations, if the proposed Mini Project adhering to the qualitative aspects mentioned above gets completed in odd semester, then that group can be allowed to work on the extension of the Mini Project with suitable improvements/modifications or a completely new project idea in even semester. This policy can be adopted on case by case basis.

Term Work	
The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of mini project to be evaluated on continuous basis, minimum two reviews in each semester.	
In continuous assessment focus shall also be on each individual student, assessment based on individual's contribution in group activity, their understanding and response to questions.	

Distribution of Term work marks for both semesters shall be as below:	Marks
1 Marks awarded by guide/supervisor based on logbook	10
2 Marks awarded by review committee	10
3 Quality of Project report	05

Review / progress monitoring committee may consider following points for assessment based on either one year or half year project as mentioned in general guidelines

One-year project:

1	In first semester entire theoretical solution shall be ready, including components/system selection and cost analysis. Two reviews will be conducted based on presentation given by students group. • First shall be for finalization of problem • Second shall be on finalization of proposed solution of problem.
2	In second semester expected work shall be procurement of component's/systems, building of working prototype, testing and validation of results based on work completed in an earlier semester. • First review is based on readiness of building working prototype to be conducted. • Second review shall be based on poster presentation cum demonstration of working model in last month of the said semester.

Half-year project:

1	In this case in one semester students' group shall complete project in all aspects including, • Identification of need/problem • Proposed final solution • Procurement of components/systems • Building prototype and testing
2	Two reviews will be conducted for continuous assessment, • First shall be for finalization of problem and proposed solution • Second shall be for implementation and testing of solution.

Assessment criteria of Mini Project.

Mini Project shall be assessed based on following criteria;	
1	Quality of survey/ need identification
2	Clarity of Problem definition based on need.
3	Innovativeness in solutions
4	Feasibility of proposed problem solutions and selection of best solution
5	Cost effectiveness
6	Societal impact
7	Innovativeness

8	Cost effectiveness and Societal impact
9	Full functioning of working model as per stated requirements
10	Effective use of skill sets
11	Effective use of standard engineering norms
12	Contribution of an individual's as member or leader
13	Clarity in written and oral communication
	In **one year, project**, first semester evaluation may be based on first six criteria's and remaining may be used for second semester evaluation of performance of students in mini project.
	In case of **half year project** all criteria's in generic may be considered for evaluation of performance of students in mini project.

Guidelines for Assessment of Mini Project Practical/Oral Examination:

1	Report should be prepared as per the guidelines issued by the University of Mumbai.
2	Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organizations having experience of more than five years approved by head of Institution.
3	Students shall be motivated to publish a paper based on the work in Conferences/students competitions.

Mini Project shall be assessed based on following points;

1	Quality of problem and Clarity
2	Innovativeness in solutions
3	Cost effectiveness and Societal impact
4	Full functioning of working model as per stated requirements
5	Effective use of skill sets
6	Effective use of standard engineering norms
7	Contribution of an individual's as member or leader
8	Clarity in written and oral communication