CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Technology & Engineering

Bachelor of Technology Programme

Information Technology

(B.Tech. IT)

ACADEMIC REGULATIONS & SYLLABUS

(Choice Based Credit System)

Faculty of Technology and Engineering

B. Tech. Programme

(Information Technology)

Chandubhai S. Patel Institute of Technology

Devang Patel Institute of Advance Technology and Research

Academic Year 2025-26

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF TECHNOLOGY AND ENGINEERING

ACADEMIC REGULATIONS
Bachelor of Technology Programmes
Choice Based Credit System

To ensure uniform system of education, duration of undergraduate and post graduate programmes, eligibility criteria for and mode of admission, credit load requirement and its distribution between course and system of examination and other related aspects, following academic rules and regulations are recommended.

1. System of Education

Choice based Credit System with Semester pattern of education shall be followed across The Charotar University of Science and Technology (CHARUSAT) both at Undergraduate and Master's levels. Each semester will be at least 90 working day duration. Every enrolled student will be required to take a course works in the chosen subject of specialization and also complete a project/dissertation if any. Apart from the Programme Core courses, provision for choosing University level electives and Programme/Institutional level electives are available under the Choice based credit system.

2. Duration of Programme

(i)	Undergraduate programme	(B. Tech.)
	Minimum	8 semesters (4 academic years)
	Maximum	16 semesters (8 academic years)

3. Eligibility for admissions

As enacted by Govt. of Gujarat from time to time.

4. Mode of admissions

As enacted by Govt. of Gujarat from time to time.

5. Programme structure and Credits

As per annexure – I attached

6. Attendance

6.1 All activities prescribed under these regulations and listed by the course faculty members in their respective course outlines are compulsory for all students pursuing the courses. No exemption will be given to any student from attendance except on account of serious personal illness or accident or family calamity that may genuinely prevent a student from attending a particular session or a few sessions. However, such unexpected absence from classes and other activities will be required to be condoned by the Dean/Principal.

6.2 Student attendance in a course should be Minimum 80%.

7 Course Evaluation

7.1 The performance of every student in each course will be evaluated as follows:

7.1.1 Internal evaluation by the course faculty member(s) based on continuous assessment, for 30% of the marks for the course; and

7.1.2 Final examination by the University through written paper or practical test or oral test or presentation by the student or a combination of any two or more of these, for 70% of the marks for the course.

7.2 *University Examination*

7.2.1 The final examination by the University for 70% of the evaluation for the course will be through written paper and 100% for practical test or oral test or presentation by the student or a combination of any two or more of these.

7.2.2 **In order to earn the credit in a course a student has to obtain grade other than FF.**

7.3 Performance at Internal & University Examination

7.3.1 Minimum performance with respect to internal marks as well as university examination will be an important consideration for passing a course. Details of minimum percentage of marks to be obtained in the examinations (internal/external) are as follows

Minimum marks in University Exam per subject	Minimum marks Overall per subject
40%	45%

7.3.2 A student failing to score 45% of the final examination will get a FF grade.

7.3.3 If a candidate obtains minimum required marks per subject but fails to obtain minimum required overall marks, he/she has to repeat the university examination till the minimum required overall marks are obtained.

8 Grading

8.1 The total of the internal evaluation marks and final University examination marks in each course will be converted to a letter grade on a ten-point scale as per the following scheme:

Table: Grading Scheme (UG)

Range of Marks (%)	≥80	≥73 ‹80	≥66 ‹73	≥60 ‹66	≥55 ‹60	≥50 ‹55	≥45 ‹50	‹45
Letter Grade	AA	AB	BB	BC	CC	CD	DD	FF
Grade Point	10	9	8	7	6	5	4	0

8.2 The student's performance in any semester will be assessed by the Semester Grade Point Average (SGPA). Similarly, his performance at the end of two or more consecutive semesters will be denoted by the Cumulative Grade Point Average (CGPA). The SGPA and CGPA are calculated as follows:

(i) SGPA = $\Sigma C_i G_i / \Sigma C_i$ where C_i is the number of credits of course i
G_i is the Grade Point for the course i
and i = 1 to n, n = number of courses in the semester

(ii) CGPA = $\Sigma C_i G_i / \Sigma C_i$ where C_i is the number of credits of course i
G_i is the Grade Point for the course i
and i = 1 to n, n = number of courses of all semesters up to which CGPA is computed.

(iii) No student will be allowed to move further if CGPA is less than 3 at the end of every academic year.

(iv) A student will not be allowed to move to third year if he/she has not cleared all the courses of first year.

(v) A student will not be allowed to move to fourth year if he/she has not cleared all the courses of second year.

9. Awards of Degree

9.1 Every student of the programme who fulfils the following criteria will be eligible for the award of the degree:

9.1.1 He should have earned at least minimum required credits as prescribed in course structure; and

9.1.2 He should have cleared all internal and external evaluation components in every course; and

9.1.3 He should have secured a minimum CGPA of 4.5 at the end of the programme;

9.1.4 In addition to above, the student has to complete the required formalities as per the regulatory bodies, if any.

9.2 The student who fails to satisfy minimum requirement of CGPA at the end of program will be allowed to improve the grades so as to secure a minimum CGPA for award of degree. Only latest grade will be considered.

10. Award of Class

The class awarded to a student in the programme is decided by the final CGPA as per the following scheme:

Distinction:	CGPA ≥ 7.5
First class:	CGPA≥ 6.0
Second Class:	CGPA≥ 5.0

11. Transcript

The transcript issued to the student at the time of leaving the University will contain a consolidated record of all the courses taken, credits earned, grades obtained, SGPA, CGPA, class obtained, etc.

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
(CHARUSAT)

FACULTY OF TECHNOLOGY & ENGINEERING

(FTE)

CHOICE BASED CREDIT SYSTEM

FOR

BACHELOR OF TECHNOLOGY & ENGINEERING

 Academic Year 2025-26

Choice Based Credit System

With the aim of incorporating the various guidelines initiated by the University Grants Commission (UGC) to bring equality, efficiency and excellence in the Higher Education System, Choice Based Credit System (CBCS) has been adopted. CBCS offers wide range of choices to students in all semesters to choose the courses based on their aptitude and career objectives. It accelerates the teaching-learning process and provides flexibility to students to opt for the courses of their choice and / or undergo additional courses to strengthen their Knowledge, Skills and Attitude.

1. CBCS – Conceptual Definitions / Key Terms (Terminologies)

1.1. Core Courses

1.1.1 University Core (UC)

University Core Courses are those courses which all students of the University of a Particular Level (PG/UG) will study irrespective of their Programme/specialisation.

1.1.2 Programme Core (PC)

A 'Core Course' is a course which acts as a fundamental or conceptual base for Chosen Specialisation of Engineering. It is mandatory for all students of a particular Programme and will not have any other choice for the same.

1.2 Elective Course (EC)

An 'Elective Course' is a course in which options / choices for course will be offered. It can either be for a Functional Course / Area or Streams of Specialization / Concentration which is / are offered or decided or declared by the University/Institute/Department (as the case may be) from time to time.

1.2.1 Institute Elective Course (IE)

Institute Courses are those courses which any students of the University/Institute of a Particular Level (PG/UG) will choose as offered or decided by the University/Institute from time-to-time irrespective of their Programme /Specialisation

1.2.2 Programme Elective Course (PE):

A 'Programme Elective Course' is a course for the specific programme in which students will opt for specific course(s) from the given set of functional course/ Area or Streams of Specialization options as offered or decided by the department from time-to-time

1.2.3 Cluster Elective Course (CE):

An 'Elective Course' is a course which students can choose from the given set of functional course/ Area or Streams of Specialization options (eg. Common Courses to EC/CE/IT/EE) as offered or decided by the Institute from time-to-time.

1.3 Non-Credit Course (NC) - AUDIT Course

A 'Non-Credit Course' is a course where students will receive Participation or Course Completion certificate. This will not be reflected in Student's Grade Sheet. Attendance and Course Assessment is compulsory for Non-Credit Courses

Charotar University of Science & Technology

Chandubhai S Patel Institute of Technology

Devang Patel Institute of Advance Technology and Research

Department of Information Technology

Vision

To become a front-runner for quality education, development and research in the field of IT.

Mission

- *To prepare next-generation technocrats for societal upliftment.*
- *To inculcate moral and ethical values for building vibrant nation.*

	CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY (CHARUSAT)												
	TEACHING & EXAMINATION SCHEME FOR B TECH PROGRAMME IN IT												
Sem	Course Code	Course Title	Teaching Scheme						Examination Scheme				
			Contact Hours				Credit		Theory		Practical		Total
			Theory	Practical	Tutorial	Project	Theory	Practical	Internal	External	Internal	External	
TY Sem -5	IT361	Software Engineering	2	4	0	0	2	2	15	35	50	50	150
	IT362	Artificial Intelligence & Machine Learning	4	2	0	0	4	1	30	70	25	25	150
	IT343	Operating System	4	2	0	0	4	1	30	70	25	25	150
	HS Elective	HS 131.02A Communication and Soft Skills/ HS 142 A French	0	2	0	0	0	2	0	0	30	70	100
		Elective - 5.1	3	2	0	0	3	1	30	70	25	25	150
	IT363	Project-II	0	0	0	2	0	1	0	0	25	25	50
	IT346	Summer Internship-I	0	0	0	3	0	3	0	0	75	75	150
	Elective 5.2	IT396 Introduction to Research Methodology/IT397 Competitive Programming Essentials	2				0	2	0	0	50	50	100
			13	**14**	**0**	**5**	**13**	**13**	**105**	**245**	**305**	**345**	**1000**
TY Sem -6	IT366	Mobile Application Development	0	4	0	0	0	4	0	0	100	100	200
	IT365	Language Processors	3	2	0	0	3	1	30	70	25	25	150
	IT348	Cryptography & Network Security	4	2	0	0	4	1	30	70	25	25	150
	IT367	Cloud Computing	3	2	0	0	3	1	30	70	25	25	150
		Elective – 6.1	3	2	0	0	3	1	30	70	25	25	150
	IT368	Project – III	0	0	0	2	0	1	0	0	25	25	50
	HS132 .02A	Contributory Personality Development	0	2	0	0	0	2	-	-	30	70	100
	Elective 6.2	IT398 Research Writing and Ethics / IT399 Mastering Competitive Programming	2				0	2	0	0	50	50	100
			13	**16**	**0**	**2**	**13**	**13**	**120**	**280**	**305**	**345**	**1050**

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY (CHARUSAT)					
LIST OF ELECTIVE SUBJECTS FOR B TECH PROGRAMME IN IT					
	Code	**Elective – 5.1**		**Code**	**Elective – 6.1**
ELECTIVES	IT389	Data Analytics & Visualization		OCIT3002	Introduction to Internet of Things (NPTEL)
	OCIT3001.01	Ethical hacking (NPTEL)		OCIT3003	Blockchain and Its Applications (NPTEL)
	IT390	Embedded Programming		OCIT3004	Deep Learning (NPTEL)
	IT391.01	Design Patterns & Frameworks			

B. Tech. (Information Technology) Programme

SYLLABI

(Semester – 5)

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

IT361: SOFTWARE ENGINEERING

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	2	4	-	6	4
Marks	50	100	-	150	

Objective of the Course:

The main objectives for offering the course Software Engineering are:

- To describe the concepts of Software requirements gathering and analyzing, Software design techniques, implementation guidelines,
- To explain CASE tools, design concepts, automated Software Testing, Documentationand Maintenance.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum Number of Hours
1	Introduction to Software and Software Engineering	04
2	Managing Software Project	06
3	Software Requirement Analysis and Design	08
4	Software Coding & Testing	08
5	Software Maintenance and Configuration Management	04

Total hours (Theory): 30

Total hours (Lab): 60

Total hours: 90

C. Detailed Syllabus:

1.	**Introduction to Software and Software Engineering**	**04 hours**	**13%**
1.1	The Evolving Role of Software		
1.2	Software: A Crisis on the Horizon and Software Myths		
1.3	Software Engineering: A Layered Technology		
1.4	Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models, Spiral Model, Agile Process Model		
2	**Managing Software Project**	**06 hours**	**20%**
2.1	Software Metrics (Process, Product and Project Metrics)		
2.2	Software Project Estimations		
2.3	Software Project Planning (MS Project & Visio Tool)		
2.4	Project Scheduling & Tracking (Earn Value Analysis)		
2.5	Risk Analysis & Management (Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation)		
3	**Software Requirement Analysis and Design**	**08 hours**	**27%**
3.1	Requirement Specification (SRS)		
3.2	Requirement Engineering		
3.3	Design Concepts and Design Principal		
3.4	Architectural Design		
3.5	Component Level Design (Function Oriented Design, Object Oriented Design) (MS Visio Tool)		
4	**Software Coding & Testing**	**08 hours**	**27%**
4.1	Coding Standard and coding Guidelines		
4.2	Code Review		
4.3	Testing Strategies		
4.4	Testing Techniques and Test Case, Test Suites Design		
5	**Software Maintenance and Configuration Management**	**04 hours**	**13%**
5.1	Types of Software Maintenance, Re-Engineering, Reverse Engineering, Forward Engineering		
5.2	The SCM Process, Identification of Objects in the Software Configuration		
5.3	Version Control and Change Control		

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will bediscussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lectures and laboratory which carries 5 Marksweightage.
- Two internal exams will be conducted and average of the same will be converted toequivalent of 15 Marks as a part of internal theory evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weightage of 5 Marks as a part of internal theory evaluation.
- Surprise tests/Quizzes/Seminar will be conducted which carries 5 Marks as a part of internal theory evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in thelaboratory.

E. Student Learning Outcome:

After completion of the course students will be able to

CO1	Prepare SRS (Software Requirement Specification) document and SPMP (Software Project Management Plan) document.
CO2	Apply the concept of Functional Oriented and Object-Oriented Approach for Software Design.
CO3	Recognize how to ensure the quality of software product, different quality standards and software review techniques.
CO4	Apply various testing techniques and test plan in.
CO5	Able to understand modern Agile Development and Service OrientedArchitecture Concept of Industry.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	1	-	-	2	2	3	2
CO2	3	3	3	1	2	1	-	-	-	-	3	2
CO3	3	2	2	2	2	-	-	-	-	2	2	2
CO4	2	3	2	3	-	-	-	-	-	-	2	2
CO5	3	3	3	3	3	-	-	-	-	-	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ Text Books:

1. Roger S.Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International Editions

❖ Reference Books:

1. Engineering Software as a Service An Agile Software Approach,Armando Fox andDavid Patterson
2. Ian Sommerville, Software engineering, Pearson education Asia
3. PankajJalote, An Integrated Approach to Software Engineering by, Springer
4. Rajib Mall, Fundamentals of software Engineering,Prentice Hall of India.
5. John M Nicolas, Project Management for Business, Engineering and Technology,Elsevier

❖ Web Materials:

1. www.en.wikipedia.org/wiki/Software_engineering
2. www.win.tue.nl
3. www.rspa.com/spi
4. www.onesmartclick.com/engsineering/software-engineering.html
5. www.sei.cmu.edus
6. https://www.edx.org/school/uc-berkeleyx

IT362: ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	4	2	-	6	5
Marks	100	50	-	150	

Objective of the Course:

The main objectives for offering the course Machine Learning are:

- To learn about the most effective machine learning techniques and gain practice.
- To able to effectively use the common neural network, including initialization, dropout regularization, Batch normalization, gradient checking.
- To understand industry best-practices for building deep learning applications.
- To learn how to quickly and powerfully apply these techniques to new problems.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction to Artificial Intelligence and Machine Learning	11
2.	Supervised Learning	11
3.	Neural Networks and Deep Learning	15
4.	Unsupervised Learning & Reinforcement Learning	12
5.	Natural Language processing and Generative AI	11

Total hours (Theory): 60
Total hours (Lab): 30
Total hours: 90

C. Detailed Syllabus:

1 Introduction to Artificial Intelligence and Machine Learning — **11 Hours** — **18 %**

The History of Artificial Intelligence, The AI Problems, AI Techniques, Applications of AI, Strong AI vs Weak AI, The Turing Test, The concept of Rationality, Ethics, and Safety of AI, Need for Machine Learning, Challenges, Types of Machine Leaning: Supervised Learning, Unsupervised Learning, Reinforcement Learning, Exploratory Data Analysis, Regression and Classification.

2 Supervised Learning — **11** — **21**

K - Nearest Neighbors, Tree based models(Decision Tree, Random Forest,

		Hours	%
	XGboost) Support Vector Machines, Regression evaluation measures (SSE, RMSE, R2 Score), Classification Evaluation measures (Accuracy, Precision, Recall , confusion Metrics), Over fitting and under fitting .		
3	**Neural Networks and Deep Learning** Perceptron Learning, Neural Network Representation, Non-Linear Activation Functions, Cost Function and Back propagation, Training & Validation, Deep Learning introduction and requirement, Hyper parameter tuning, Convolution Neural Nets, Recurrent Neural Nets.	**15 Hours**	**20 %**
4	**Unsupervised Learning & Reinforcement Learning** Clustering and Anomaly Detection, Dimensionality Reduction (PCA, SVD), Reinforcement Learning fundamentals and Applications.	**12 Hours**	**20%**
5	**Natural Language processing and Generative AI** History of NLP , Key NLP Tasks, Text Preprocessing , Machine Learning in NLP , NLP Libraries and Tools , NLP Applications , State-of-the-Art Models, Introduction to Generative AI, Introduction to Generative Adversarial Networks and Style Transfer, State-of-the-Art Generative Models.	**11 Hours**	**21%**

D. Instructional Method and Pedagogy:

At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.

- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lectures and laboratory which carries 5 Marks weightage.
- Two internal exams will be conducted and average of the same will be converted to equivalent of 15 Marks as a part of internal theory evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weightage of 5 Marks as a part of internal theory evaluation.
- Surprise tests/Quizzes/Seminar will be conducted which carries 5 Marks as a part of internal theory evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in the laboratory.

E. Student Learning Outcome:

Upon completion of this course, students will be able:

1. Understand both the statistical and machine learning terminology necessary to create a foundation for understanding the similarity between both the streams.
2. Appreciate the underlying mathematical relationships within and across Machine

Learning algorithms and the paradigms of supervised learning.

3. Compare the strengths and weaknesses of many popular Neural Network and Deep learning based machine learning models.
4. Design and evaluate the unsupervised models to solve complex real world problems.
5. Design and develop the code for recommender system, apply various reinforcement algorithms and apply it to solve real world problems.

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	-	2	-	1	1	-	1	-	1	1	2
CO2	2	1	-	2	-	-	-	-	-	-	-	-	1	1
CO3	2	-	3	2	2	-	-	-	-	2	-	1	3	2
CO4	2	3	2	2	-	-	-	-	-	-	-	-	1	1
CO5	2	3	2	-	2	-	-	-	-	2	-	1	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ Text Books:

1. Machine Learning, Tom Mitchell, McGraw Hill, 1997. ISBN 0070428077
2. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, 2004
3. "Generative Deep Learning: Teaching Machines To Paint, Write, Compose, and Play", Shroff/O'Reilly, by David Foster

❖ Reference Books:

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.
2. Richard O. Duda, Peter E. Hart & David G. Stork, "Pattern Classification. Second Edition", Wiley & Sons, 2001.
3. Trevor Hastie, Robert Tibshirani and Jerome Friedman, "The elements of statistical learning", Springer, 2001.
4. Richard S. Sutton and Andrew G. Barto, "Reinforcement learning: An introduction", MIT Press, 1998.

❖ Web Materials:

1. https://www.youtube.com/watch?v=fgtUFzxNztA
2. http://nptel.iitm.ac.in/video.php?courseId=1041
3. http://www-formal.stanford.edu/jmc/whatisai/whatisai.html
4. http://www.webopedia.com/TERM/A/artificial_intelligence.html
5. http://en.wikipedia.org/wiki/Artificial_intelligence

IT343: OPERATING SYSTEM

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	4	2	0	6	5
Marks	100	50	0	150	

Objective of the Course:

The main objectives for offering the course Operating System are:

- The primary objective of this course is to provide students with a comprehensive understanding of the fundamental principles and design considerations of modern operating systems.
- It aims to familiarize students with key OS concepts such as process management, memory management, file systems, input/output systems, and system security.
- The course also emphasizes the importance of resource allocation, scheduling, concurrency, and system-level programming, preparing students to analyze and evaluate the performance and structure of various operating systems in real-world environments.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction	03
2.	System Structures	03
3.	Process Management	06
4.	Process scheduling	06
5.	Process Coordination	06
6.	Deadlocks	06
7.	Memory Management	15
8.	File System	06
9.	Secondary Storage Structure & I/O systems	06
10.	System Security	03
	Total hours (Theory):	60
	Total hours (Lab):	30
	Total hours:	90

C. Detailed Syllabus:

1.	**Introduction** What is Operating System & evolution of OS, Computer-System Organization & Architecture, OS Structure & Operations, Special purpose Systems, Open-source OS	**03 hours**	**05 %**
2.	**System Structures** OS Services, System calls, Types of system calls, OS Structure: Layered, Microkernel, Operating system Generation, Booting	**03 hours**	**05 %**
3.	**Process Management** Process, Process Control Block, Process States, Scheduling concepts, Process creation Threads, Types of Threads, Multithreading, Issues & termination	**06 hours**	**10 %**
4.	**Process scheduling** Concept, Scheduler, Preemptive Scheduling, Criteria, Scheduling Algorithms: FCFS, SJF, RR, Priority, Multi-queue	**06 Hours**	**10 %**
5.	**Process Coordination** Race Conditions, Critical Section, Peterson's Solution, Hardware Solution, Strict Alternation, Semaphores Classical IPC Problems: TheBounded-Buffer (Producer Consumer) Problem, Reader's & Writer Problem, Dinning Philosopher Problem, Monitors	**06 hours**	**10 %**
6.	**Deadlocks** Deadlock Problem, Deadlock Characterization, Resource- allocation graph, Deadlock Prevention, Deadlock avoidance: RAG & Banker's algorithm for single & multiple resources, Deadlock Detection, Recovery	**06 hours**	**10 %**
7.	**Memory Management** Address binding, Address space, Swapping, Contiguous MemoryAllocation Paging, Page table: Hierarchical, Hashed, Inverted Segmentation, Virtual-Memory: DemandPaging, PageReplacement algorithms: FIFO, Optimal, LRU, second chance, LFU & MFU, Working set model, Thrashing, Frame Allocation	**15 hours**	**25 %**
8.	**File System** File concept, Access methods, Directory & Disk Structure, File protection: Type, access control File SystemStructure, Implementation, Directory Implementation, Allocation Methods, Free space management,	**06 hours**	**10 %**
9.	**Secondary Storage Structure & I/O systems** Disk: structure, Arm scheduling: FCFS, SSTF, SCAN, LOOK, Formatting & Boot block, RAID Structure & levels I/O Hardware, Interrupt, DMA, Block & Character devices, Network devices, Transforming I/O request to Hardware Operations	**06 hours**	**10 %**
10	**System Security** Goals of protection, domain of protection, Trojan Horse, Viruses, Worms	**03 hours**	**05 %**

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lectures and laboratory which carries 5 Marks weightage.
- Two internal exams will be conducted and average of the same will be converted to equivalent of 15 Marks as a part of internal theory evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weightage of 5 Marks as a part of internal theory evaluation. Surprise tests/Quizzes/Seminar will be conducted which carries 5 Marks as a partof internal theory evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in the laboratory.

E. Student Learning Outcome:

Upon completion of this course, students will be able to do the following:

CO1	Students will able to exhibit familiarity with the fundamental concepts ofoperating systems
CO2	Students will able to exhibit competence in recognizing operating systemsfeatures and issues
CO3	Students will able to apply a mature understanding of operating system designed how it impacts application systems design and performance.

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	-	-	2	2
CO2	2	3	2	-	1	-	-	-	-	-	-	-	3	2
CO3	2	2	3	1	1	-	-	-	-	-	-	-	2	3

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ **Text Books:**

1. Operating System Concepts, 9th Edition by Avi Silberschatz, Peter Baer Galvin,Greg Gagne, Wiley Publication.

❖ **Reference Books:**

1. Modern Operating Systems, 3rd Edition by Andrew S. Tanenbaum, PHI

2. Operating System – Internals & Design Principles, William Stallings, PHI
3. Operating Systems, D. M. Dhamdhare, TMH
4. Unix System Concepts & Applications, 4E, Sumitabha Das, TM
5. UNIX Shell Programming, Yashwant Kanitkar, BPB Publications.

❖ **Web Materials:**

1. NPTEL course: https://onlinecourses.nptel.ac.in/noc25_cs141/preview

IT363: PROJECT - II

A. Credits and Hours:

Teaching Scheme	Theory	Project	Tutorial	Total	Credit
Hours/week	0	2	-	2	1
Marks	0	50	-	50	

Objective of the Course:

The main objectives for offering the course Project-II are:

- To motivate students to develop innovative, practical solutions aligned with the UN Sustainable Development Goals (SDGs).
- To provide a platform for applying academic knowledge to real-world challenges through structured project development.
- To foster creativity, teamwork, and problem-solving through ideation, prototyping, and presentation stages.
- To encourage participation in research, startup initiatives, and external competitions for holistic development.
- To enhance communication and technical skills through pitching and poster presentations.

B. Outline of the Course:

- Student at the beginning of a semester may be advised by his/her supervisor (s) for recommended courses.
- Students are required to get approval of project definition from the department.
- After approval of project definition students are required to report their project work weekly to respective internal guide.
- Project will be evaluated at least once per week in laboratory during the semesterand final submission at the end of the semester as a part of continuous evaluation.
- Project work should include whole SDLC of development of software / hardwaresystem as a solution of particular problem by applying principles of Software Engineering.
- Students have to submit project with following listed documents at the time of final submission.
    a. Project Synopsis
    b. Software Requirement Specification
    c. SPMP

d. Final Project Report/paper

e. Project Setup file with Source code [Uploaded on GitHub]

f. Project Presentation (PPT)

g. Video Recording (Per Project) A student has to produce some usefuloutcome by conducting experiments or project work.

- A student has to produce some useful outcome by conducting experiments or project work.

Total hours (Theory): 00
Total hours (Lab): 30
Total hours: 30

C. Instructional Method and Pedagogy:

1. Project Groups would be form of maximum two students.
2. Inter batch group formation is not permitted due to difficulties in progress tracking.
3. Students are advised to choose innovative and challenging definitions.
4. Batch wise project definitions must be unique.
5. Any management system would not be encouraged.
6. Tools like GitHub would be used to track the progress of project development by theconcern faculty. Concerned guide will demonstrate the working of GitHub Tool.
7. Student has to prepare report at end of semester as part of submission.
8. Report structure is finalized for semester end submission.
9. To have a better outcome as well as progress tracking at the end of semester, it is decided that students have to appear for two internal reviews, which will help them to get more insight in the project.
10. To maintain similarity below 40%, Students have to submit project's final document to concern SGP guide for plagiarism check before 15 days of external exam.
11. Students have to attach plagiarism report in final spiral bound with duly signed by SGP guide.
12. Students have to bring internal review card hard copy on the day of internal review exam, after that they will bring filled review card on the day of external review.

D. Student Learning Outcome:

After the completion of the course students will able to

CO1	Identify a range of solutions, critically evaluate and justify proposed design solution.
CO2	Manage learning & self-development including development of organizational skills, time management, effective use of scientific literature and discriminating use of Web resources.
CO3	Apply a wide range of principles and tools available to the software developer such as choice of the algorithm, language, software libraries etc.
CO4	Write and test programs using appropriate test cases.
CO5	Solve communication issues in large, complex software projects and Structure & Communicate ideas effectively orally. Also Prepare & deliver coherent and structuredverbal and written technical reports.
CO6	Evaluate system in terms of general quality attributes and possible trade-offs presented within the given problem/system.

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	2	-	2	2	2	2	-	1	3	3	2
CO2	3	3	1	2	1	2	2	2	2	-	1	3	3	2
CO3	3	1	3	3	3	1	2	2	3	-	2	3	3	2
CO4	3	1	1	3	1	-	-	1	2	-	2	2	2	2
CO5	3	-	-	-	-	-	2	3	3	3	3	2	2	1
CO6	3	2	1	2	1	-	-	1	2	-	1	1	3	1

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

E. Recommended Study Material:

❖ Reference book:

1. John M Nicolas, Project Management for Business, Engineering and Technology,Elsevier.
2. Sanjay Mohapatra, Software Project Management, Cengage Learning
3. Clive L. Dym, Patrick Little, Elizabeth J. Orwin, "Engineering Design – A ProjectBased Introduction", Wiley India Pvt. Ltd.
4. B. Hughes & M. Cotterell, "Software Project Management", Tata Mcgraw Hills.

❖ Web Materials:

1. https://status.net/templates/project-report/
2. https://www.tutorialspoint.com/software_engineering/software_project_management.htm
3. https://www.geeksforgeeks.org/coding-standards-and-guidelines/
4. https://www.altexsoft.com/blog/engineering/8-ways-to-improve-software-testing-through-planning-work-environment-automated-testing-and-reporting/

5. https://nptel.ac.in/courses/106/105/106105218/
6. https://www.youtube.com/watch?v=T3q6QcCQZQg
7. https://www.scrib
8. br.com/category/research-paper/

IT346: SUMMER INTERNSHIP-I

A. Credit and Hours:

Teaching Scheme	Project		Total	Credit
	Internal	External		
Hours	45	45	90	3
Marks	75	75	150	

B. Instructional Method and Pedagogy:

- Summer internship shall be at least 90 hours during the summer vacation only.
- Department/Institute will help students to find an appropriatecompany/industry/organization for the summer internship.
- The student must fill up and get approved a Summer Internship Acceptance form bythe company and provide it to the Coordinator of the department within the specifieddeadline.
- Students shall commence the internship after the approval of the department Coordinator. Summer internships in research centers is also allowed.
- During the entire period of internship, the student shall obey the rules and regulationsof the company/industry/organization and those of the University.
- Due to inevitable reasons, if the student will not able to attend the internship for fewdays with the permission of the supervisor, the department Coordinator should be informed via e-mail and these days should be compensated later.
- The student shall submit two documents to the Coordinator for the evaluation of thesummer internship:
- Summer Internship Report
- Summer Internship Assessment Form
- Upon the completion of summer internship, a hard copy of "Summer Internship Report" must be submitted through the presentation to the Coordinator by the first day of the new term.
- The report must outline the experience and observations gained through practical internship, in accordance with the required content and the format described in this guideline. Each report will be evaluated by a faculty member of the department on a satisfactory/unsatisfactory basis at the beginning of the semester.
- If the evaluation of the report is unsatisfactory, it shall be returned to the student forrevision and/or rewriting. If the revised report is still unsatisfactory the student shallbe requested to repeat the summer internship.

C. Format of Summer Internship Report:

The report shall comply with the summer internship program principles. Main headings are to be centered and written in capital boldface letters. Sub-titles shall be written in small letters and boldface. The typeface shall be Times New Roman font with 12pt. All the margins shall be 2.5cm. The report shall be submitted in printed form and filed. An electronic copy of the report shall be recorded in a CD and enclosed in the report. Each report shall be bound in a simple wire vinyl file and contain the following sections:

- Cover Page
- Page of Approval and Grading
- Abstract page: An abstract gives the essence of the report (usually less than one page). Abstract is written after the report is completed. It must contain the purpose and scope of internship, the actual work done in the plant, and conclusions arrived at.
- TABLE OF CONTENTS (with the corresponding page numbers)
- LIST OF FIGURES AND TABLES (with the corresponding page numbers)
- DESCRIPTION OF THE COMPANY/INDUSTRY/ORGANISATION: Summarize the work type, administrative structure, number of employees (how many engineers, under which division, etc.), etc. Provide information regarding
    - Location and spread of the company
    - Number of employees, engineers, technicians, administrators in the company
    - Divisions of the company
    - Your group and division
    - Administrative tree (if available)
    - Main functions of the company
    - Customer profile and market share
- INTRODUCTION: In this section, give the purpose of the summer internship, reasons for choosing the location and company, and general information regarding the nature of work you carried out.
- PROBLEM STATEMENT: What is the problem you are solving, and what are the reasons and causes of this problem.
- SOLUTION: In this section, describe what you did and what you observed during the summer internship. It is very important that majority of what you write should be based on what you did and observed that truly belongs to the company/industry/organization.
- CONCLUSIONS: In the last section, summarize the summer internship activities. Present your observations, contributions and intellectual benefits. If this is your

second summer internship, compare the first and second summer internships and your preferences.

- REFERENCES: List any source you have used in the document including books, articles and web sites in a consistent format.
- APPENDICES: If you have supplementary material (not appropriate for the main body of the report), you can place them here. These could be schematics, algorithms, drawings, etc. If the document is a datasheet and it can be easily accessed from the internet, then you can refer to it with the appropriate internet link and document number. In this manner you don't have to print it and waste tons of paper.

D. Learning Outcomes:

At the end of the course, the students will be able to

CO1	Demonstrate ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
CO2	Cultivate an understanding of their multidisciplinary interest, including theskills, responsibilities and career path of professionals through practice-oriented and 'hands-on' working experience.
CO3	An exhibit foresight, independent thinking, resourcefulness, and the ability to make decisions.
CO4	Develop a right work attitude, self-confidence, interpersonal skills and ability to work as a team in an industry.

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	3	-	-	-	-	-	-	-	-	3	2
CO2	1	2	3	2	-	-	-	-	-	-	-	-	1	1
CO3	1	1	2	-	-	-	-	2	2	3	-	-	2	1
CO4	-	-	-	-	-	-	-	-	-	-	-	-	1	3

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

IT396: INTRODUCTION TO RESEARCH METHODOLOGY (ELECTIVE – 5.2)

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	-	2	-	2	**2**
Marks	-	100	-	100	

Pre-requisite courses:

Basic Mathematics and Statistics, Subject-Specific Knowledge, Computer Literacy.

Objective of the Course:

The main objectives for offering the course Introduction to Research Methodology are:

- Enable students to create research designs and effectively collect, analyze and interpret data using appropriate statistical tools and software.
- Enable students to create sound research designs and effectively collect, analyze, and interpret data using appropriate statistical tools and software.
- Enhance the academic and technical writing skills with attention to style, structure, and citation standards.
- Familiarize students with modern tools such as reference managers (Zotero, Mendeley), and document preparation systems (LaTeX, MS Word, Overleaf) to support efficient research documentation and formatting.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction to Research	06
2.	Literature Review	10
3.	Technical Writing	08
4.	Tools / Techniques for Research	06
	Total hours (Theory) :	-
	Total hours (Lab) :	30
	Total hours :	30

C. Detailed Syllabus:

1. Introduction to Research — **06 Hours** — **25%**

Meaning, Objective and Motivation in Research. Types of Research: Descriptive vs. Analytical, Applied vs. Fundamental, Quantitative vs. Qualitative, Conceptual vs. Empirical, Concept of Applied and Basic Research Process, Criteria of Good Research.

2. Research Design, Data Collection and Analysis — **08 Hours** — **30%**

Features of Good Design, Types of Research Design, Basic Principles of Experimental Design. Primary and Secondary Data, Methods of Data Collection, Sampling Methods, Data Processing and Analysis Strategies and Tools, Data Analysis with Statistical Packages

3. Literature Review in in the respective areas of Specialization — **10 Hours** — **30%**

Objectives of Review of Literature, Importance of Literature Review in Defining a Problem, Steps in Conducting Literature Review - Guidelines on Style, Mechanics and language usage - Writing up to the literature Reviewed - Some Examples

4. Tools / Techniques for Research — **06 Hours** — **15%**

Methods to search required information effectively, Reference Management Software like Zotero/ Mendeley, Software for paper formatting like LaTeX/MS Office, Overleaf.

D. Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Explain the fundamental concepts and principles of research methodology, including the scientific method and various research paradigms.
CO2	Design a comprehensive research plan, including selecting appropriate research strategies, methodologies, and data collection techniques..
CO3	Differentiate between and apply quantitative and qualitative research methods appropriately based on the research question.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	1	-	-	-	-	-	-	-	-	-	1	-
CO2	1	2	1	-	-	-	-	-	-	-	-	-	1	-
CO3	1	2	2	-	-	-	-	-	-	-	-	-	1	-

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

Recommended Study Material:

- **Text Books:**
  1. C.R. Kothari and Gaurav Garg, “Research Methodology: Methods and Techniques”, New Age International (P) Ltd., Publishers, Fourth Multi Colour Edition, 2020.
- **Reference Books:**
  1. Research Methodology And Statistical Tools - P. Narayana Reddy And G.V,R.K, Acharyulu, l" Edition, Excel Books, New Delhi, 2008

IT397: COMPETITIVE PROGRAMMING ESSENTIALS (ELECTIVE – 5.2)

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	-	2	0	2	**2**
Marks	-	100	0	100	

Pre-requisite courses:

- Data Structures and Algorithms
- Familiarity with a programming language

Objectives of the Course:

The main objectives for offering the course Introduction to Research Methodology are:

- The primary objective of this course is to equip students with a comprehensive understanding of advanced algorithmic techniques essential for competitive programming and placement preparation.
- By mastering these techniques, students will not only enhance their problem-solving skills but also gain a competitive edge in coding competitions such as the ICPC.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction to competitive programming and techniques	02
2.	Time Efficiency	04
3.	Mathematics for competitive programming	04
4.	Linear data structures and Libraries	05
5.	Non-linear data structures and Libraries	05
6.	Sorting, searching and algorithm design	06
7.	String processing algorithms	04
	Total hours (Theory) :	00
	Total hours (Lab) :	30
	Total hours :	30

C. Detailed Syllabus:

1. Introduction to competitive programming and techniques **02 Hours** **07%**

Meaning Of Competitive Programming, Programming Techniques (Language Features Input/Output, Numbers, Code Shortening), Recursive Algorithms

(Generating Subsets, Generating Permutations, Backtracking), Bit Manipulation (Bit Operations, Set Operations).

2.	**Time Efficiency** Common Time Complexity and Calculation Rules, Examples (Maximum Sub-Array Sum Problem, 2 Queens), Algorithm Analysis, Situational Awareness, Touch Typing Exercise to Improve Typing Speed.	**04 Hours**	**13%**
3.	**Mathematics for competitive programming** Geometry (Points and Lines), Closest Pair Problem, Number Theory, Sum Formulae and Logarithms, Inclusion and Exclusion, Combinatorics	**04 Hours**	**13%**
4.	**Linear data structures and Libraries** Static Array, Dynamic Array, Boolean Array, LinkedList, Stack, Queue, Double-Ended Queue, Exercise Problems.	**05 Hours**	**17%**
5.	**Non-linear data structures and Libraries** Binary Search Tree (Map/Set), Heap (Priority Queue), Hash Table (Unordered Map), Graph (Adjacency List and Matrix, Edge List), Exercise Problems.	**05 Hours**	**17%**
6.	**Sorting, searching and algorithm design** Complete Search (Iterative and Recursive), Linear and Binary Search, Sorting Algorithm and Library, Two Pointers, Nearest Smaller/Greater Element, Sliding Window, Ternary Search, Minimizing Sum, hamming distance, counting sub grids, reachability in graphs	**06 Hours**	**20%**
7.	**String processing algorithms** Basic string processing, string processing problem categories, string matching, longest common subsequence, levenshtein distance	**04 Hours**	**13%**

D. Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Apply advanced algorithmic techniques to solve complex problems.
CO2	Develop the ability to analyze and optimize the time complexity of algorithms.
CO3	Gain mathematical skills crucial for algorithmic problem-solving
CO4	Design efficient algorithms and develop strong problem-solving skills.
CO5	Compete in competitive programming contests and coding interviews.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	-	1	-	-	-	-	-	-	-	-	2	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-	3	-
CO3	2	1	-	-	-	-	-	-	-	-	-	-	2	-
CO4	2	2	-	-	-	-	-	-	-	-	-	-	3	-
CO5	3	3	-	1	-	-	-	-	-	-	-	-	2	-

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

E. Recommended Study Material:

- **Text book:**
    1. Competitive programming by Steven & Felix Halim
    2. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests.

IT389: DATA ANALYTICS & VISUALIZATION (ELECTIVE – 5.1)

A. Credit Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

Objective of the Course:

The main objectives for offering the course Data Analytics & Visualization are:

- Apply statistical methods to perform exploratory data analysis (EDA) and gain insights from complex datasets.
- Enable students to create sound research designs and effectively collect, analyze, and interpret data using appropriate statistical tools and software.
- Enhance the academic and technical writing skills with attention to style, structure, and citation standards.
- Familiarize students with modern tools such as reference managers (Zotero, Mendeley), and document preparation systems (LaTeX, MS Word, Overleaf) to support efficient research documentation and formatting.

B. Outline of the course:

Sr No.	Title of the unit	Minimum number of hours
1.	Intro to data visualization & data analysis	5
2	Data Visualizations and Storytelling in Python	10
3.	Data Visualization and EDA using PowerBI	15
4.	Charting toolbox in Tablue	10
5.	Data Manipulation with Pandas	5

Total hours (Theory): 45Hrs.
Total hours (Lab): 30Hrs.
Total hours: 75 Hrs

C. Detailed Syllabus:

1. Intro to data visualization & data analysis **5Hrs** **11%**

- Getting to Know a Dataset
- Data Cleaning and Imputation
- Relationships in Data
- Visualizing distributions
- Visualizing two variables

- Avoid the most common plot problems

2.	**Data Visualizations and Storytelling in Python**	**10Hrs**	**22%**

- Exploratory Analysis
- Introduction to Seaborn
- Visualizing Two Quantitative Variables
- Visualizing a Categorical and a Quantitative Variable
- Customizing Seaborn Plots
- Storytelling using Seaborn

3.	**Data Visualization and EDA using PowerBI**	**15Hrs**	**33%**

- Intro to PowerBI
- Transforming Data
- Visualizing Data
- Filtering
- Reducing Cognitive Load
- Initial Exploratory Data Analysis in Power BI
- Distributions and Outliers
- EDA with Categorical Variables
- Relationships between Continuous Variables

4.	**Charting toolbox in Tablue**	**10 Hrs**	**22%**

- Getting Started with Tableau
- Building and Customizing Visualizations
- Presenting Your Data
- Best Practices in Data Visualization
- Maps and Spatial Visualizations
- Charting Toolbox

5.	**Data Manipulation with Pandas**	**5Hrs**	**12%**

- Transforming Data Frames
- Aggregating Data Frames
- Slicing and Indexing Data Frames
- Creating and Visualizing Data Frames

D. Instructional Method and Pedagogy:

- At the start of the course, the course delivery pattern, and prerequisite of the subject will be discussed.
- Faculty would use a coached problem-solving method as it is the class format in which faculty provide a structured, guided context for students working collaboratively to solve problems.
- Attendance is compulsory in lectures and laboratory which qualifying criteria for overall evaluation.
- Minimum two internal exams will be conducted.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval.
- Surprise tests/Quizzes/Theory assignments will be part of overall evaluation.
- Minimum 10 experiments are suggested in the laboratory related to course content.

E. Student Learning Outcome:

After learning the course, students will able to

CO1	Create data visualizations and apply statistical methods to investigate data
CO2	Develop working knowledge of Python language for analyzing data using Python libraries
CO3	Examine, navigate, and learn to use the various features of Tableau for story telling using data
CO4	Create interactive exploratory analyses, reports, and dashboards using Power BI
CO5	Differentiate between different data roles such as Data Engineer, Data Analyst, Data Scientist, Business Analyst

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO01	PSO02
CO1	2	1	-	2	1	-	-	-	-	1	-	1	1	1
CO2	1	2	-	1	2	-	-	-	-	2	-	1	1	1
CO3	-	-	-	1	3	-	-	-	-	3	-	1	1	1
CO4	-	1	-	-	3	-	-	-	-	3	-	1	1	1
CO5	-	1	-	-	-	1	-	-	-	-	-	3	1	1

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ **Text Books:**

1. Making Data Visual: A Practical Guide to Using Visualization for Insight (Grayscale edition)

❖ **Reference Books:**

1. Data Analysis and Visualization Using Python: Analyze Data To Create Visualizations For Bi Systems by Embarak, Apress
2. Learning Microsoft Power BI Transforming Data into Insights, by O'Reilly
3. Visual Data Storytelling with Tableau 2018 Edition by Lindy Ryan, PEARSON INDIA
4. Communicating Data with Tableau - Designing, Developing, and Delivering Data Visualizations (Covers Tableau version 8.1) 1 Edition (English, Paperback, Ben Jones)

❖ **Online Courses:**

1. https://www.coursera.org/professional-certificates/google-data-analytics
2. https://www.coursera.org/learn/data-driven-decisions-with-power-bi
3. https://www.coursera.org/specializations/data-visualization?=

IT390: EMBEDDED PROGRAMMING (ELECTIVE – 5.1)

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

Objective of the Course:

The main objectives for offering the course Embedded Systems are:

- To have a basic proficiency in a traditional embedded C language including an ability to write simple to intermediate programs and an ability to understand code written in that language.
- To have an understanding of the basic issues of embedded software development and associated hardware.
- To have a basic understanding of some of the more advanced topics of embedded systems.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum Number of hours
1.	Introduction to Embedded System.	05
2.	Embedded Software.	10
3.	Embedded System Development.	10
4.	Interface Digital and Analog I/O Devices (Arduino Interfacing)	05
5.	Real Time Operating System.	10
6.	Case Study of embedded and real-time operating systems, real time applications	05

Total hours (Theory):45

Total hours (Lab): 30

Total hours: 75

C. Detailed Syllabus:

1. Introduction to Embedded System. 05 Hours 13%

1.1 Characteristics of Embedded System.

1.2 Types of Embedded Systems.

1.3 Examples of Embedded Systems.

2. Embedded Software. **10 Hours** **35%**

2.1 Embedded Programming in C and C++

2.2 Source Code Engineering Tools for Embedded C/C++

2.3 Program Modeling Concepts in Single and Multiprocessor Systems

2.4 Software Development Process

2.5 Software Engineering Practices in the Embedded Software Development

3. Embedded System Development. **10 Hours** **20%**

3.1 Embedded software development tools – Emulators and debuggers.

3.2 Design issues and techniques

3.3 Case studies

3.4 Complete design of example embedded systems

4. Interface Digital and Analog I/O Devices (Arduino Interfacing): **05 Hours** **17%**

4.1 Basic Interfacing and I/O Concept Interfacing LED

4.2 Switch,7seg LED its and Code Interfacing POT, LM35, Accelerometer (ADXL3C5C) and its Code

4.3 Interfacing keypad, Initialization for serial port, Interfacing DC motor, Interfacing 16x2 LCD

5. Real-Time Operating Systems **10 Hours** **20%**

5.1 Typical OS structure.

5.2 RTOS structure.

5.3 The context of its use.

5.4 Schedule management for multiple tasks.

5.5 Scheduling in real time.

5.6 Interrupt routines in RTOS environment.

5.7 RTOS task scheduling models.

5.8 List of basic actions in pre-emptive scheduler and expected time taken.

6. Embedded system Applications (Arduino): **05 Hours** **06%**

6.1 Case study of RTOS using MUCOS.

6.2 Case study for RTOS based programming.

6.3 Coding for Automatic Chocolate vending machine using MUCOS.

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Understand the fundamental skills knowledge of embedded system, different examples, its characteristics
CO2	Understand the different processors modeling concepts in single and multiprocessor Systems.
CO3	Analyse the requirement of interfacing peripheral devices with Microcontroller
CO4	Understand operating system and do basic programming of real time operating system
CO5	Analyse the systems requirements to meet the specifications

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	-	-	-	-	1	-	-	-	2	-	-
CO2	3	2	-	-	-	-	-	-	2	1	-	1	1	-
CO3	3	3	2	-	1	-	-	-	2	1	-	1	1	-
CO4	3	1	1	-	1	-	-	-	2	1	-	1	1	-
CO5	3	3	3	2	3	3	2	2	3	1	1	2	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lectures and laboratory which carries 5 Marks weightage.
- Two internal exams will be conducted and average of the same will be converted to equivalent of 15 Marks as a part of internal theory evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weightage of 5 Marks as a part of internal theory evaluation.
- Surprise tests/Quizzes/Seminar will be conducted which carries 5 Marks as a part of internal theory evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in the laboratory.

E. Student Learning Outcome:

Upon completion of this course, students will be able:

- To solve difficult and complex problem of computer science using Embedded Systems
- To select any R&D field related to application of Embedded Systems in PG courses.
- To develop hardware based solutions to industry problems
- To develop software solution as per need of today's IT edge which requires high automation and less human intervention.

F. Recommended Study Material:

❖ Text Books

1. Rajkamal, "Embedded System: Architecture, Programming and Design" Tata McGraw-Hill, 2003.
2. WayneWolf, “Computers as Components: Principles of Embedded Computer SystemDesign”, Elsevier, 2006.

❖ Reference Books

1. SriramIyer and Pankaj Gupta, “Embedded Real Time Systems Programming”, Tata McGraw-Hill, 2004.
2. F. Vahid, T. Givargis, Embedded System Design, John Wiley and Sons, 2002
3. Code generation for Embedded Processors by Peter Marwedel, G. Goosens, Kluner Academic Pub. 1993.
4. An Embedded Software Primer Addition, by David E. Simon ,Wesely.

OCIT3001.01: ETHICAL HACKING (ELECTIVE – 5.1)

Description:

This course OCIT3001 - Ethical Hacking is offered from SWAYAM as noc22_cs13 – Ethical Hacking

Credit and Week:

Teaching Scheme	Week	Marks	Credit
	4/8/12	100	2/3/4/5

About the course:

Ethical hacking is a subject that has become very important in present-day context, and can help individuals and organizations to adopt safe practices and usage of their IT infrastructure. Starting from the basic topics like networking, network security and cryptography, the course will cover various attacks and vulnerabilities and ways to secure them. There will be hands-on demonstrations that will be helpful to the participants. The participants are encouraged to try and replicate the demonstration experiments that will be discussed as part of the course.

Pre-requisites:

1. Basic concepts in programming and networking

Industry support:

TCS, Wipro, CTS, Google, Microsoft, Qualcomm

A. Course layout:

Week 1: Introduction to ethical hacking. Fundamentals of computer networking. TCP/IP protocol stack.
Week 2: IP addressing and routing. TCP and UDP. IP subnets.
Week 3: Routing protocols. IP version 6.
Week-4: Installation of attacker and victim system. Information gathering using advanced google search, archive.org, netcraft, whois, host, dig, dnsenum and NMAP tool.
Week-5: Vulnerability scanning using NMAP and Nessus. Creating a secure hacking environment. System Hacking: password cracking, privilege escalation, application execution. Malware and Virus. ARP spoofing and MAC attack.
Week 6: Introduction to cryptography, private-key encryption, public-key encryption.
Week 7: Cryptographic hash functions, digital signature and certificate, applications.
Week 8: Steganography, biometric authentication, network-based attacks, DNS and Email security.
Week-9: Packet sniffing using Wireshark and burpsuite, password attack using burp suite. Social engineering attacks and Denial of service attacks.
Week 10: Elements of hardware security: side-channel attacks, physical inclinable functions, hardware trojans.
Week-11: Different types of attacks using Metasploit framework: password cracking, privilege escalation, remote code execution, etc. Attack on web servers: password attack, SQL injection, cross site scripting.

Week 12: Case studies: various attacks scenarios and their remedies.

B. Couse Outcome (CO):

At the end of the course, the students will be able to

CO1	Understanding of computer networking principles, including TCP/IP protocol stack and IP addressing.
CO2	Exploring advanced techniques for gathering information using tools like NMAPand Nessus, and understand the importance of creating a secure hacking environment.
CO3	Learning cryptographic concepts such as private-key and public-key encryption, cryptographic hash functions, and their practical applications.
CO4	Acquire skills in system hacking, including password cracking, privilege escalation, and malware execution, along with countermeasures.
CO5	Developing proficiency in various ethical hacking techniques, including ARPspoofing, social engineering, and exploiting vulnerabilities using tools like Metasploit, with an emphasis on real-world case studies and remedies.

C. Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO 11**	**PO12**	**PSO1**	**PSO2**
CO1	3	-	-	-	-	-	1	-	-	-	-	-	1	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-	-	-
CO3	2	1	-	2	1	-	-	-	-	-	-	-	2	-
CO4	3	2	2	2	2	-	-	-	-	-	-	-	2	-
CO5	3	-	3	-	-	-	-	-	-	-	2	2	3	2

Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

If there is no correlation, put "-"

D. Books and References

Text Books / Basic Material

1. Data and Computer Communications -- W. Stallings.
2. Data Communication and Networking -- B. A. Forouzan
3. TCP/IP Protocol Suite -- B. A. Forouzan
4. UNIX Network Programming -- W. R. Stallings
5. Introduction to Computer Networks and Cybersecurity -- C-H. Wu and J. D. Irwin
6. Cryptography and Network Security: Principles and Practice -- W. Stallings
7. ** Various web resources **.

Criteria to get a certificate:

- Average assignment score = 25% of average of best 8 assignments out of the total 12 assignments given in the course.
- Exam score = 75% of the proctored certification exam score out of 100
- Final score = Average assignment score + Exam score
- ***YOU WILL BE ELIGIBLE FOR A CERTIFICATE ONLY IF AVERAGE ASSIGNMENT SCORE >=10/25 AND EXAM SCORE >= 30/75. If one of the 2 criteria is not met, you will not get the certificate even if the Final score >= 40/100.***

IT391.01 DESIGN PATTERNS & FRAMEWORKS (ELECTIVE – 5.1)

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

Objective of the Course:

The main objectives for offering the course are:

- Understanding Software Design Principles
- Exploring Various Design Patterns and Application
- Practically Applying and gain insight of quality code
- Exploring Software Frameworks to Leverage Design Patterns
- Exploring Industry Best Practices for better software development

B. Outline of the Course:

Sr. No.	Title of the Unit	Minimum Number of Hours
1.	Introduction to Design Patterns	05
2.	Creational Design Patterns	08
3.	Structural Design Patterns	08
4.	Behavioral Design Patterns	08
5.	Application of Design Patterns in Software Frameworks	16

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

C. Detailed Syllabus:

1. Introduction to Design Patterns **05 Hours** **11.11%**

What are Design Patterns?

Object-Oriented Design Principles

Types of Design Patterns (Creational, Structural, Behavioral)

Benefits of Using Design Patterns

2.	**Creational Design Patterns** Understanding and working with various Creational Design Patterns :Singleton Pattern, Factory Method Pattern, Abstract ,Factory Pattern, Builder Pattern, Prototype Pattern	**08 Hours**	**17.77%**
3.	**Structural Design Patterns** Understanding and working with various Structural Design Patterns: Adapter Pattern, Bridge Pattern, Composite Pattern, Decorator Pattern, Facade Pattern, Proxy Pattern	**08 Hours**	**17.77%**
4.	**Behavioral Design Patterns** Understanding and working with various Behavioral Design Patterns: Observer Pattern, Strategy Pattern, Command Pattern State Pattern, Template Method Pattern, Chain of Responsibility Pattern, Visitor Pattern	**08 hours**	**17.77%**
5.	**Application of Design Patterns in Software Frameworks** Understanding the Concept of Software Frameworks Exploring various Frameworks and Comparing Working with .net core Framework for building modern app Working with python based Framework: Django	**16 hours**	**35.55%**

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board etc.
- Attendance is compulsory in lectures and laboratory.
- Marks will be given based on continues evaluation, i.e. Unit Tests/Surprise tests/Quizzes/Projects/Presentation and Assignments based on course content will be given to the students at the end of each unit/topic.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in the laboratory.

E. Student Learning Outcome Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Understand Design Principles
CO2	Apply and Identify Design Patterns
CO3	Create Maintainable Software

CO4	Explore Various Software Frameworks
CO5	Analyze Real-World Case Studies

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	-	-	-	-	-	-	-	-	-	-	1	-
CO2	3	2	2	2	-	-	-	-	-	-	-	-	2	-
CO3	3	-	2	-	3	-	-	-	-	-	2	2	3	-
CO4	2	2	-	-	-	-	-	-	-	-	-	-	2	-
CO5	2	2	-	2	-	-	-	-	-	-	-	2	3	-

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

- **Books**
  1. Design Patterns: Elements of Reusable Object-Oriented Software, by Gamma Erich (Author), Addison-Wesley Professional.
  2. Design Patterns in .NET: Reusable Approaches in C# and F# for Object-Oriented Software Design, by Dmitri Nesteruk, Apress.
  3. Django Design Patterns and Best Practices - Second Edition: Industry-standard web development techniques and solutions using Python" by by Arun Ravindran, Ingram short title.
- **Reference Links/ e-content:**
  1. https://www.dotnettricks.com/learn/designpatterns/gang-of-four-gof-design-patterns-in-net
  2. https://github.com/DovAmir/awesome-design-patterns
  3. https://docs.djangoproject.com/en/4.2/

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	--	02	--	30	**02**
Marks	**--**	100	--	100	

Pre-requisite courses:

- Communicative English

Objectives of the Course:

- To hone and sharpen Communication Skills of students
- To prepare globally and multi-culturally competent communicators and professionally compatible cadre of future professionals
- To equip and empower students to qualify and successfully clear all the phases of selection procedure for on and off campus interviews

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	An Introduction to Communication	06
2.	Cross-cultural Communication and Globalization	03
3.	Communication for Career Building	10
4.	Group Dynamics and Soft Skills	05
5.	Effective Presentation Strategies	04
6.	Contemporary Issues in Communication and Soft Skills	02
	Total hours (Theory) :	--
	Total hours (Practical) :	**30**
	Total hours :	**30**

C. Detailed Syllabus:

1.	**An Introduction to Communication** Basics of Communication: Origin, Concept, Process, Levels, Principles and Barriers; Applications of Communication; Rhetoric in Professional Communication; Importance of Ethos, Logos, and Pathos in Communication	**06 Hours**	**20%**
2.	**Cross-cultural Communication and Globalization** Basic Concepts: Culture, Globalization and Cross-cultural Communication; Social and People Skills; Communicating with People of Different Cultures; Conflicts in Cross-cultural Communication and Tactics / techniques to resolve them; Persuasive Communication	**03 Hours**	**10%**

3.	**Communication for Career Building** Cover Letters and Resume; E-mail and Report; Types of Resume; Concept and Rationale of Group Discussion Skills and Aspects assessed in Group Discussion; Concept and Rationale of Personal Interview; Types of Personal Interview; Writing Statement of Purpose	**10 Hours**	**33%**
4.	**Group Dynamics and Soft Skills** An Introduction to Group Dynamics and Soft Skills; Groups and their Structures; Roles and Functions of Members in Groups; Conflict Management; Aptitude and Attitude; Various Intelligences; Developing an Open Mindset	**05 Hours**	**17%**
5.	**Effective Presentation Strategies** Designing Appealing Presentation; Audience Analysis and Supporting Material; Presentation Mechanics and Presentation Process; Managing Yourself during Q and A Session; Fundamentals of Persuasion	**04 Hours**	**14%**
6.	**Contemporary Issues in Communication and Soft Skills** Trends and Practices in Communication, Case Studies	**02 Hours**	**06%**

D. Recommended Study Material:

❖ **Text book:**

1. Koneru, A. Professional Communication, Tata McGrow Hill Education Private Limited
2. Disanza, J.R. &Legge, N. Business and Professional Communication, Pearson Education
3. Raman, M & Singh, P. Business Communication, Oxford University Press

❖ **Reference book:**

1. Disanza, J.R. &Legge, N. Business and Professional Communication, Pearson Education
2. Anandamurugan, A. Placement Interviews – Skills for Success, Tata McGrow Hill Education Private Limited

❖ **Web material:**

1. https://www.coursera.org/learn/careerdevelopment
2. https://www.futurelearn.com/courses/writing-applications
3. https://www.futurelearn.com/courses/workplace-englis

SYLLABI

(Semester – 6)

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

IT366: MOBILE APPLICATION DEVELOPMENT

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	-	4	-	4	4
Marks	-	200	-	200	

Objective of the Course:

The main objectives for offering the course Mobile Application Development are:

- To introduce students to the fundamentals of mobile platforms, app development lifecycle, and mobile user experience (UX) design principles.
- To provide hands-on experience in building mobile applications using Flutter framework and Dart programming language.
- To explain core concepts of Dart including syntax, data types, functions, object-oriented programming, and state management.
- To develop proficiency in designing and implementing responsive user interfaces using built-in and custom widgets in Flutter.
- To demonstrate integration of backend services using Firebase, including authentication, real-time database, and advanced Flutter features like animations and debugging.
- To expose students to native Android app development using Kotlin, focusing on language fundamentals, object-oriented features, and app structure.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction to Mobile App Development	04
2.	Flutter Basics	10
3.	Dart Programming	12
4.	Flutter User Interface (Flutter UI)	18
5.	Firebase Integration & Advanced Topics	08
6.	Native Application development using Kotlin(Android)	08
	Total hours (Theory) :	**00**
	Total hours (Lab) :	**60**
	Total hours :	**60**

C. Detailed Syllabus:

1.	**Introduction to Mobile App Development** Overview of mobile platforms, mobile app development lifecycle, User experience (UX) considerations for mobile apps	**02 Hours**	**7%**
2.	**Flutter Basics** Introduction to Flutter and Dart, Flutter architecture and widgets, Comparison with other mobile app development frameworks, Components of flutter, Building user interfaces with Flutter	**05 Hours**	**17%**
3.	**Dart Programming** Introduction to Dart programming language, Importing and using a library, Creating dart libraries, dart packages, Packages structure, Dart syntax, Dart data types, Dart functions, Dart classes and objects, Dart state management- built-in state management (setState)	**06 Hours**	**28%**
4.	**Flutter User Interface (Flutter UI)** Introduction to Flutter, flutter compilation & rendering, widgets introduction, hello flutter, Understanding built-in widgets- Layout widgets, Styling widgets, Stateful widgets, Stateless widgets, Creating UI with widgets, creating custom widgets, Handling user gestures, validating input, custom input and form field, Navigation and Routing in Flutter.	**09 Hours**	**28%**
5.	**Firebase Integration & Advanced Topics** Introduction to Firebase platform, Firebase Authentication for user login and registration, Cloud Firestore for real-time data storage, Advanced animations and transitions in Flutter, Testing and Debugging app.	**04 Hours**	**14%**
6.	**Native Application development using Kotlin(Android)** Introduction, Why Kotlin? Data types, Functions, Looping & Ranges, Expression & statements, Objects Everywhere, Creating Class, Constructors, Packages, Exceptions, Property Accessors, overloading, Enumerations, Data Classes, Self-Calls & the Elvis Operator, Introduction to Generics.	**04 Hours**	**7%**

D. Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Create visually appealing and responsive UIs using Flutter widgets and styling techniques.
CO2	Apply state management solutions to handle user interactions and maintain app state.
CO3	Develop Android-specific features using the Kotlin programming language.
CO4	Test and debug mobile applications to ensure functionality and performance.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	2	1	2	-	-	-	-	-	-	-	1	-
CO2	3	2	3	1	3	-	-	-	-	-	-	-	2	-
CO3	2	2	2	2	3	-	-	-	-	-	-	-	1	-
CO4	1	2	2	1	3	-	-	-	-	-	-	-	1	-

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

E. Recommended Study Material:

- **Text book:**
  1. Bruce Eckel and Svetlana Isakova, Atomic Kotlin, Mindview LLC.
  2. Alessandro Biessek, Flutter for Beginners, Packt Publishing Limited

- **Reference book:**
  1. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips, Chris Stewart, and Kristin Marsicano.
  2. "iOS Programming: The Big Nerd Ranch Guide" by Christian Keur and Aaron Hillegass.
  3. "Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart" by Alessandro Biessek.

- **Web material:**
  1. https://kotlinlang.org/
  2. https://flutter.dev/

- **Software:**
  1. Android Studio
  2. Xcode
  3. Visual Studio

IT365: LANGUAGE PROCESSORS

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Total	Credit
Hours/week	3	2	5	4
Marks	100	50	150	

Objective of the Course:

The main objectives for offering the course Language Processor are:

- To study Language processor and language processing activities.
- To explore design and implement lexical analyzer and parser.
- To explore, design code generation schemes.
- To explore optimization of codes.
- To learn the assembly language processing

B. Outline of the Course:

Sr No.	Title of the unit	Minimum number of hours
1.	Introduction to Language Processor	04
2.	Macros and Macro Preprocessors	02
3.	Finite Automata and Grammar	10
4.	Analysis Phase of Compiler	15
5.	Synthesis Phase of Compiler	10
6.	Assemblers	04

Total hours (Theory): 45 Hrs.

Total hours (Lab): 30 Hrs.

Total hours: 75 Hrs.

C. Detailed Syllabus:

1. Introduction to Language Processor **04 hours** **07 %**

1.1 Introduction

1.2 Language processing activities

1.3 Fundamental of language processing

1.4 Fundamental of language Specification

1.5 Introduction to preprocessor, compiler and assembler

2.	**Macros and Macro Preprocessors**	**02 hours**	**03 %**
2.1	Macro definition and call		
2.2	Macro Expansion, Nested Macro Calls		
3.	**Finite Automata and Grammar**	**10 Hours**	**17 %**
3.1	Basic Definition, Regular Expression, Regular Language, Finite Automata : NFA and DFA		
3.2	Non Determinism Finite Automata, Conversion from NFA to DFA		
3.3	$\wedge$- Non Determinism Finite Automata, Conversion of NFA- $\wedge$ to NFA		
3.4	Minimization of DFA		
3.5	Introduction to Grammar, Types of Grammars		
3.6	Context Free Grammars, Derivations and Languages, Relationship between derivation and derivation trees		
3.7	Ambiguity Unambiguous CFG and Algebraic Expressions Bacos Naur Form (BNF), Normal Form – CNF, GNF		
4.	**Analysis Phase of Compiler**	**15 hours**	**25 %**
4.1	Introduction to Lexical analysis, Role of the lexical analyzer		
4.2	Specification of tokens, Recognition of tokens		
4.3	Lexical analyzer generators		
4.4	Role of the parser		
4.5	Top-down parsing, Bottom- up parsing		
4.6	Syntax-Directed Definitions		
4.7	Bottom-Up Evaluation of S-Attributed Definitions and L-Attributed Definitions		
4.8	Top Down Translation and Bottom-Up Evaluation of Inherited Attributes		
5.	**Synthesis Phase of Compiler**	**10 hours**	**17 %**
5.1	Intermediate Languages, Declarations, Assignment Statements, Intermediate code generation techniques		
5.2	The Principal Sources of Optimization		
5.3	Machine Independent and machine dependent code optimization techniques		
5.4	Issues in the Design of a Code Generator		
6.	**Assemblers**	**04 hours**	**06 %**
6.1	Elements of assembly language programming		
6.2	Overview of the assembly process		
6.3	A simple Assembly Scheme		
6.4	Design of two pass assembler		

D. Instructional Method and Pedagogy:

- Lectures will be taken in class room with the use of multi-media presentations and black board – mix of both.

- Assignments based on above course content will be given to the students at the end of each chapter. Each assignment contains minimum 5 questions.
- Quizzes and Surprise tests will be conducted for testing the knowledge of students for particular topic.

E. Student Learning Outcome:

At the end of the course, the students will be able to

CO1	Analyze the functionalities of language processors.
CO2	Simulate Compilation process using tools such as LEX and YACC.
CO3	Analyze and generate the different parsing techniques.
CO4	Perform optimization at different level of program.

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	-	3	-	-	-	-	-	-	-	3	2
CO2	3	3	3	2	3	-	-	-	-	-	-	2	3	1
CO3	2	3	2	2	-	-	-	-	-	-	-	-	3	2
CO4	3	3	2	3	-	-	-	-	-	-	-	-	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ **Text Books:**

1. Alfred Aho, Ravi Sethi, Jeffrey D Ullman, "Compilers Principles, Techniques and Tools", Pearson Education Asia.
2. D. M. Dhamdhere, "System Programming and Operating Systems", Tata McGraw-Hill.
3. John c martin, "Introduction to Languages and the Theory of Computation", The McGraw -Hill.

❖ **Reference Books:**

1. Allen I. Holub "Compiler Design in C", Prentice Hall of India.
2. C. N. Fischer and R. J. LeBlanc, "Crafting a compiler with C", Benjamin Cummings.
3. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill
4. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI.
5. Kenneth C. Louden, "Compiler Construction: Principles and Practice", Thompson Learning.
6. Compiler Construction by Kenneth. C. Louden, Vikas Pub.

❖ **Web Materials:**

1. http://compilers.iecc.com/crenshaw
2. http://www.compilerconnection.com
3. http://dinosaur.compilertools.net
4. http://pltplp.net/lex-yacc

IT348: CRYPTOGRAPHY & NETWORK SECURITY

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	4	2	0	6	5
Marks	100	50	0	150	

Objective of the Course:

The main objectives for offering the course Cryptography & network Security are:

- To introduce cryptography theories, algorithms and systems. Necessary approaches and techniques to build protection mechanisms in order to secure computer networks.
- To explain the basics of modern cryptography including symmetric key cryptography, public key cryptography, secure hash and digital signature.
- To explore issues surrounding secure key management, random number generation, and the incorporation of cryptography into legacy applications.
- To analyze performance of various cryptographic and cryptanalytic algorithms.

B. Outline of the Course:

Sr No.	Title of the unit	Minimum number of hours
1.	Introduction and Mathematical Foundations	09
2.	Symmetric Key Ciphers	15
3.	Public Key Cryptography	09
4.	Message Authentication and HashFunction	09
5.	Network Security	12
6.	System Security	06
	Total hours (Theory)	60
	Total hours (Lab)	30
	Total hours	90

C. Detailed Syllabus:

1. Introduction and Mathematical Foundations 09 hours 15 %

1.1 Security trends – Attacks, Services and Mechanism

1.2	Conventional Encryption Model, Classical Encryption Techniques, Different types of ciphers, Steganography		
1.3	Basic Number theory—Prime And Relative Prime Numbers, Modular Arithmetic, Congruence ,Fermat and Euler's theorem, Euclid's Algorithm, Chinese Remainder theorem, LFSR sequences, Finite fields.		
2.	**Symmetric Key Ciphers**	**15 hours**	**25 %**
2.1	Simplified Data Encryption Standard, DES, Triple DES		
2.2	Block Cipher Principles, Characteristics Of Advanced Symmetric Block Cipher, Differential And Linear cryptanalysis, Block Cipher Design Principles		
2.3	Advanced Encryption Standard Algorithm,RC4 and RC5		
2.4	Modes of Operation		
2.5	Pseudorandom Number generator and function, Key Distribution		
3.	**Public Key Cryptography**	**09 hours**	**15%**
3.1	Principles Of Public-Key Cryptography		
3.2	RSA Algorithm		
3.3	Key Management		
3.4	ElGamal Algorithm		
3.5	Diffie-Hellman Key Exchange		
4.	**Message Authentication and Hash Function**	**09 hours**	**15 %**
4.1	Authentication Requirement		
4.2	Hash Functions ,Message Authentication Code, Security Of Hash Functions And MAC		
4.3	MD5 Message Digest Algorithm, Secure Hash Algorithm, HMAC		
4.4	Authentication protocols ,Digital Signatures, DSS,		
5.	**Network Security**	**12 hours**	**20%**
5.1	Authentication Applications—Kerberos, X.509 Directory Authentication Service,		
5.2	Electronic Mail Security—PGP ,S/MIME		
5.3	IP security —Overview, ESP, AH, Transport and Tunnel mode in IP Sec		
5.4	Web Security— Web Security Requirement, SSL, TLS,SET		

6. System Security **06 hours** **10%**

6.1 Intruders, Viruses and Related Threats

6.2 Firewall Design Principles

6.3 Trusted Systems

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will bediscussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHPetc.
- Attendance is compulsory in lectures and laboratory which carries 5 Marks.
- Two internal exams will be conducted and average of the same will be converted to equivalent of 15 Marks as a part of internal theory evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weightage of 5 Marks as a part of internal theory evaluation.
- Surprise tests/Quizzes/Seminar will be conducted which carries 5 Marks as a part of internal theory evaluation.
- The course includes a laboratory, where students have an opportunity to build anappreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in the laboratory.

E. Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Know the importance of security and to apply the concepts of techniques andmethods to implement security mechanism.
CO2	Learn the different encryption and decryption algorithms using symmetric &asymmetric approach to provide confidentiality.
CO3	Implements the aspects of integrity and authentication, like digital signature andmessage digest, and map them with practical use of it.
CO4	To learn the concepts of web application security, network security and systemsecurity for making them immune to attack.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	-	1	-	1	1	-	-	-	1	2
CO2	3	3	3	2	3	1	-	-	-	-	-	-	1	1
CO3	2	2	3	2	3	2	-	-	2	1	-	-	1	1
CO4	3	3	3	2	3	-	2	1	2	1	1	2	3	3

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ Text Books:

1. William Stallings, Cryptography and Network Principles and Practice, PrenticeHall, Pearson Education Asia

❖ Reference Books:

1. Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill Companies
2. AtulKahate, Cryptography & Network Security, The McGraw-Hill Companies
3. William Stallings Network Security Essentials: Applications And Standards, Prentice Hall, Pearson Education

❖ Reference Links/ e-content:

1. http://people.csail.mit.edu/rivest/crypto-security.html
2. http://www.cryptix.org/
3. http://www.cryptocd.org/
4. http://www.cryptopp.com/

IT367: CLOUD COMPUTING

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Total	Credit
Hours/week	3	2	5	4
Marks	100	50	150	

Objective of the Course:

The main objectives for offering the course Advanced Computing are:

- To provide an overview of the basic concepts of cluster computing, grid computing and cloud computing.
- To highlight the advantage of deploying cluster computing and cloud computing.
- To illustrate the practical adoption of a cloud and cluster deployment through real life case studies.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Fundamentals of Distributed Computing	06
2.	Understanding Cloud Computing Concepts	06
3.	Cloud Enabling Technologies	05
4.	Cloud Services Providers	04
5.	Understanding and Implementing Cloud Securities	04
6.	Cloud Computing : Cost Metrics ,QoS and SLA	04
7.	Fundamentals of Container Technology & Tools	08
8.	Fundamentals of Micro services and Automation Tools	08

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

C. Detailed Syllabus:

1. Fundamentals Distributed Computing **06 hours** **14 %**

1.1 History of Computing , Elements of Distributed Computing, Parallel Computing

1.2 Scalable Parallel Computer Architecture, Symmetric Multi-Processor

1.3 Cluster Computing , Architecture and Applications

1.4 Load Balancing in Cluster Computing.

1.5 Resource Management and Scheduling in Cluster Computing

1.6	Programming Environments and Tools : Cluster Computing		
1.7	Setting up the Cluster , Monitor & security		
1.8	Implementing RPC and Web-services		
1.9	Grid Computing and Architecture		
2.	**Understanding Cloud Computing Concepts**	**06 hours**	**14 %**
2.1	History of cloud computing,		
2.2	Technology Innovations: Clustering, Grid ,Utility & Virtualization		
2.3	Cloud characteristics		
2.4	Cloud delivery Models & Deployment Models		
2.5	Cloud Storage , Virtual Private Cloud		
2.6	Challenges of Cloud Computing		
3.	**Cloud Enabling Technologies**	**05 hours**	**12 %**
3.1	Data Center Technology		
3.2	Virtualization Technology		
3.3	Case Study of Cloud Enabling Technologies		
4.	**Cloud Services Providers**	**04 hours**	**10 %**
4.1	Deploying and Accessing cloud services		
4.2	Securing Cloud Services		
4.3	Comparing Cloud Service Providers		
4.4	Amazon Web services, Google Cloud Platform ,		
4.5	Microsoft Azure, Salesforce etc.		
5.	**Understanding and Implementing Cloud Securities**	**04 hours**	**10 %**
5.1	Basic Terms: Confidentiality , Integrity, authenticity, Availability, Risk, Threat		
5.2	Cloud Security Threats		
5.3	Cloud Security Mechanisms		
5.4	Case Studies: AWS (Cloud Security)		
6.	**Cloud Computing : Cost Metrics ,QoS and SLA**	**04 hours**	**08%**
6.1	Cost Metrics : Network, Computing , Storage		
6.2	QoS(Quality of Service) and QoS Metrics , SLA (Service Level Agreement)		
7.	**Fundamentals of Container Technology & Tools**	**08 hours**	**16 %**
7.1	Understanding Basic Terms : Cgroups, Namespace, Layered File System etc.		
7.2	Understanding & Implementing Container.		

7.3 Virtual Machine vs Containers

7.4 Pros and Cons of Container Technology

7.5 Fundamentals of Docker.

7.6 Docker networking and storage

7.7 Docker Compose

7.8 Introduction to Container Orchestration and Tool: Kubernets

8. Fundamentals of Micro services and Automation Tools **08 hours** **16 %**

8.1 Introduction to Micro Services and need of Micro Services

8.2 Micro Services Architecture and Concepts/Components

8.3 Pros and Cons/Challenges and Applications of Micro Services

8.4 Introduction to DevOps and CI/CD

8.5 Introduction to Ansible : Infrastructure/Platform Automation

8.6 Introduction to Jenkins : CI/CD Automation

D. Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lectures and laboratory which carries 5 Marks weightage.
- Two internal exams will be conducted and average of the same will be converted to equivalent of 15 Marks as a part of internal theory evaluation.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval. It carries a weightage of 5 Marks as a part of internal theory evaluation.
- Surprise tests/Quizzes/Seminar will be conducted which carries 5 Marks as a part of internal theory evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments/Tutorials related to course content will be carried out in the laboratory.

E. Student Learning Outcome:

At the end of the course, the students will be able to

CO1	Understand and learn the concept of Grid, Cluster and Cloud Computing.
CO2	To solve the problems using Grid, Cluster and Cloud Computing.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	3	1	3	-	-	-	1	-	1	2	3	3
CO2	2	3	3	1	3	-	-	-	1	-	2	3	3	3

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

Text Books:

1. Erl, Thomas, Ricardo Puttini, and Zaigham Mahmood. Cloud computing: concepts, technology & architecture. Pearson Education, 2013.
2. Judith, S. Hurwitz. Cloud Computing For Dummies. John Wiley & Sons, 2019.
3. Mastering Cloud Computing. Rajkumar Buya.

Reference Books:

1. Ronald Krutz, Cloud Security, Wiley India.
2. Bernard Golden, Virualization for Dummies, Wiley India.

Web Materials:

1. www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf
2. https://cloud.google.com/gcp/
3. https://aws.amazon.com/
4. https://azure.microsoft.com/en-us/

IT368: PROJECT - III

A. Credit and Hours:

Teaching Scheme	Theory	Practical	Project	Total	Credit
Hours/week	0	0	2	2	1
Marks	0	0	50	50	

Objective of the Course:

The main objectives for offering the course Project-III are:

- To motivate students to develop innovative, practical solutions aligned with the UN Sustainable Development Goals (SDGs).
- To provide a platform for applying academic knowledge to real-world challenges through structured project development.
- To foster creativity, teamwork, and problem-solving through ideation, prototyping, and presentation stages.
- To encourage participation in research, startup initiatives, and external competitions for holistic development.
- To enhance communication and technical skills through pitching and poster presentations.

B. Outline of the Course:

Sr. No.	Title	Minimum Number of Hours
1	Software Project Planning and Tracking tools	10
2	Software Designing Tools	10
3	Software Testing Tools	10
	Total hours (Theory):	**00**
	Total hours (Lab):	**30**
	Total hours:	**30**

C. Detailed Syllabus:

1. Software Project Planning and Tracking Tools **20 Hours**

1.1 Pert Chart, Gantt Chart, MS Project and Visio

1.2 Primavera for project tracking

2. Software Project Designing Tools **20 Hours**

2.1 MS Visio, Rational Rose, Edraw Max

3. Software Testing Tools **20 Hours**

3.1 Win runner, HP Load Runner

D. Instructional Method and Pedagogy:

- Project Groups would be form of maximum two students.
- Inter batch group formation is not permitted due to difficulties in progress tracking.
- Students are advised to choose innovative and challenging definitions.
- Batch wise project definitions must be unique.
- Project based on Web development, E-commerce etc. are restricted. As they would be covered as part of curriculum in other courses.
- Tools like GitHub would be used to track the progress of project development by the concern faculty. Concerned guide will demonstrate the working of GitHub Tool.
- Student has to prepare report at end of semester as part of submission.
- Report structure is finalized for semester end submission.
- To have a better outcome as well as progress tracking at the end of semester, it is decided that students have to appear for two internal reviews, which will help them to get more insight in the project.
- To maintain similarity below 40%, Students have to submit project's final document to concern SGP guide for plagiarism check (iThenticate/ Turnitin report) before 15 days of external exam.
- Students have to attach plagiarism report in final spiral bound with duly signed by SGP guide.
- Students have to bring internal review card hard copy on the day of internal review exam, after that they will attach filled review card in their final project report.

E. Student Learning Outcome:

After the completion of the course students will able to

CO1	Identify a range of solutions, critically evaluate and justify proposed design solution.
CO2	Manage learning & self-development including development of organizational skills, time management, effective use of scientific literature and discriminating use of Webresources.
CO3	Apply a wide range of principles and tools available to the software developer such as choice of the algorithm, language, software libraries etc.
CO4	Write and test programs using appropriate test cases.
CO5	Solve communication issues in large, complex software projects and Structure &communicate ideas effectively orally. Also Prepare & deliver coherent and structured verbal and written technical reports.
CO6	Evaluate system in terms of general quality attributes and possible trade-offs presented within the given problem/system.

Course Articulation Matrix:

	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	2	-	2	2	2	2	-	1	3	3	2
CO2	3	3	1	2	1	2	2	2	2	-	1	3	3	2
CO3	3	1	3	3	3	1	2	2	3	-	2	3	3	2
CO4	3	1	1	3	1	-	-	1	2	-	2	2	2	2
CO5	3	-	-	-	-	-	2	3	3	3	3	2	2	1
CO6	3	2	1	2	1	-	-	1	2	-	1	1	3	1

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

F. Recommended Study Material:

❖ **Reference book:**

1. John M Nicolas, Project Management for Business, Engineering and Technology, Elsevier.
2. Sanjay Mohapatra, Software Project Management, Cengage Learning
3. Clive L. Dym, Patrick Little, Elizabeth J. Orwin, "Engineering Design – A Project Based Introduction", Wiley India Pvt. Ltd.
4. B. Hughes & M. Cotterell, "Software Project Management", Tata Mcgraw Hills.

❖ **Web Materials:**

1. https://status.net/templates/project-report/
2. https://www.tutorialspoint.com/software_engineering/software_project_managem e nt.htm
3. https://www.geeksforgeeks.org/coding-standards-and-guidelines/
4. https://www.altexsoft.com/blog/engineering/8-ways-to-improve-software-testing-through-planning-work-environment-automated-testing-and-reporting/
5. https://nptel.ac.in/courses/106/105/106105218/
6. https://www.youtube.com/watch?v=T3q6QcCQZQg
7. https://www.scribbr.com/category/research-paper/

OCIT3004: DEEP LEARNING (ELECTIVE – 6.1)

Description:

This course OCIT3004 - Deep Learning is offered from SWAYAM as noc25_cs21– Deep Learning.

A. Credit and Week:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

About the course:

The automatic analysis and understanding of images and videos, a field called Computer Vision, occupies significant importance in applications including security, healthcare, entertainment, mobility, etc. The recent success of deep learning methods has revolutionized the field of computer vision, making new developments increasingly closer to deployment that benefits end users. This course will introduce the students to traditional computer vision topics, before presenting deep learning methods for computer vision. The course will cover basics as well as recent advancements in these areas, which will help the student learn the basics as well as become proficient in applying these methods to real-world applications. The course assumes that the student has already completed a full course in machine learning, and some introduction to deep learning preferably, and will build on these topics focusing on computer vision.

B. Pre-requisites:

1. Completion of a basic course in Machine Learning
2. (Recommended, but not mandatory) Completion of a course in Deep Learning, or exposure to topics in neural networks
3. Knowledge of basics in probability, linear algebra, and calculus
4. Experience of programming, preferably in Python

Industry support:

All companies that use computer vision for their products/services (Microsoft, Google, Facebook, Apple, TCS, Cognizant, L&T, etc.)

C. Course layout:

Week 1: Introduction and Overview:

- Course Overview and Motivation; Introduction to Image Formation, Capture and Representation; Linear Filtering, Correlation, Convolution

Week 2: Visual Features and Representations:

- Edge, Blobs, Corner Detection; Scale Space and Scale Selection; SIFT, SURF; HoG, LBP, etc.

Week 3: Visual Matching:

- Bag-of-words, VLAD; RANSAC, Hough transform; Pyramid Matching; Optical Flow

Week 4: Deep Learning Review:

- ❖ Review of Deep Learning, Multi-layer Perceptrons, Backpropagation

Week 5: Convolutional Neural Networks (CNNs):

- ❖ Introduction to CNNs; Evolution of CNN Architectures: AlexNet, ZFNet, VGG, InceptionNets, ResNets, DenseNets

Week 6: Visualization and Understanding CNNs:

- ❖ Visualization of Kernels; Backprop-to-image/Deconvolution Methods; Deep Dream, Hallucination, Neural Style Transfer; CAM, Grad-CAM, Grad-CAM++; Recent Methods (IG, Segment-IG, SmoothGrad)

Week 7: CNNs for Recognition, Verification, Detection, Segmentation:

- ❖ CNNs for Recognition and Verification (Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss); CNNs for Detection: Background of Object Detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, RetinaNet; CNNs for Segmentation: FCN, SegNet, U-Net, Mask-RCNN

Week 8: Recurrent Neural Networks (RNNs):

- ❖ Review of RNNs; CNN + RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition

Week 9: Attention Models:

- ❖ Introduction to Attention Models in Vision; Vision and Language: Image Captioning, Visual QA, Visual Dialog; Spatial Transformers; Transformer Networks

Week 10: Deep Generative Models:

- ❖ Review of (Popular) Deep Generative Models: GANs, VAEs; Other Generative Models: PixelRNNs, NADE, Normalizing Flows, etc

Week 11: Variants and Applications of Generative Models in Vision:

- ❖ Applications: Image Editing, Inpainting, Superresolution, 3D Object Generation, Security; Variants: CycleGANs, Progressive GANs, StackGANs, Pix2Pix, etc

Week 12: Recent Trends:

- ❖ Zero-shot, One-shot, Few-shot Learning; Self-supervised Learning; Reinforcement Learning in Vision; Other Recent Topics and Applications

Course Outcome:

After completion of the course, Students will be able to

CO1	Understand the core principles of deep learning, from Bayesian learning to neural network architectures.
CO2	Gain hands-on expertise in implementing diverse deep learning algorithms like CNNs and LSTMs.
CO3	Explore unsupervised learning techniques and generative models to tackle tasks like image denoising and generation.
CO4	Master advanced optimization methods and training strategies for efficient and effective deep learning model training.
CO5	Acquire practical skills to apply deep learning in various domains such as computer vision and natural language processing for tasks like object detection and semantic segmentation.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	-	-	-	1	-	-	-	-	-	1	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-	-	-
CO3	2	1	-	2	1	-	-	-	-	-	-	-	2	-
CO4	3	2	2	2	2	-	-	-	-	-	-	-	2	-
CO5	3	-	3	-	-	-	-	-	-	-	2	2	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

D. Books and references

❖ **Text Books / Basic Material**

References for deep learning:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016
2. Michael Nielsen, Neural Networks and Deep Learning, 2016
3. Yoshua Bengio, Learning Deep Architectures for AI, 2009

References for computer vision:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, 2010.
2. Simon Prince, Computer Vision: Models, Learning, and Inference, 2012.
3. David Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2002.

Other useful references:

1. Bishop, Christopher. Neural Networks for Pattern Recognition. New York, NY: Oxford University Press, 1995. ISBN: 9780198538646.
2. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006. ISBN 978-0-387-31073-2
3. Duda, Richard, Peter Hart, and David Stork. Pattern Classification. 2nd ed. New York, NY: Wiley-Interscience, 2000. ISBN: 9780471056690.
4. Mitchell, Tom. Machine Learning. New York, NY: McGraw-Hill, 1997. ISBN: 9780070428072.
5. Richard Hartley, Andrew Zisserman, Multiple View Geometry in Computer Vision, 2004.
6. David Marr, Vision, 1982.

OCIT3002: INTRODUCTION TO INTERNET OF THINGS
(ELECTIVE – 6.1)

Description:

This course OCIT3002 - Introduction to Internet of Things is offered from SWAYAM as noc22_cs53 – Introduction to Internet of Things

A. Credit and Week:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

B. About the course:

Internet of Things (IoT) is presently a hot technology worldwide. Government, academia, and industry are involved in different aspects of research, implementation, and business with IoT. IoT cuts across different application domain verticals ranging from civilian to defense sectors. These domains include agriculture, space, healthcare, manufacturing, construction, water, and mining, which are presently transitioning their legacy infrastructure to support IoT. Today it is possible to envision pervasive connectivity, storage, and computation, which, in turn, gives rise to building different IoT solutions. IoT-based applications such as innovative shopping system, infrastructure management in both urban and rural areas, remote health monitoring and emergency notification systems, and transportation systems, are gradually relying on IoT based systems. Therefore, it is very important to learn the fundamentals of this emerging technology.

Pre-requisites:

1. Basic programming knowledge

C. Course layout:

Week 1: Introduction to IoT: Part I, Part II, Sensing, Actuation, Basics of Networking: Part-I

Week 2: Basics of Networking: Part-II, Part III, Part IV, Communication Protocols: Part I, Part II

Week 3: Communication Protocols: Part III, Part IV, Part V, Sensor Networks: Part I, Part II

Week 4: Sensor Networks: Part III, Part IV, Part V, Part VI, Machine-to-Machine Communications

Week 5: Interoperability in IoT, Introduction to Arduino Programming: Part I, Part II, Integration of Sensors and Actuators with Arduino: Part I, Part II

Week 6: Introduction to Python programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi

Week 7: Implementation of IoT with Raspberry Pi (contd), Introduction to SDN, SDN for IoT

Week 8: SDN for IoT (contd), Data Handling and Analytics, Cloud Computing

Week 9: Cloud computing (contd), Sensor-Cloud

Week 10: Fog Computing, Smart Cities and Smart Homes

Week 11: Connected Vehicles, Smart Grid, Industrial IoT

Week 12: Industrial IoT (contd), Case Study: Agriculture, Healthcare, Activity Monitoring

Course Outcome:

After completion of the course, Students will be able to

CO1	Understanding IoT Fundamentals and Networking Basics.
CO2	Mastering Communication Protocols and Sensor Networks.
CO3	Proficiency in Arduino and Python Programming for IoT Applications.
CO4	Implementing IoT Solutions with Raspberry Pi and SDN.
CO5	Exploring Advanced IoT Applications and Integration Techniques.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	-	-	1	-	-	2	1	-	-	-	2	-
CO2	-	2	-	-	-	2	1	1	1	-	2	-	1	2
CO3	1	3	2	1	-	2	-	-	2	-	1	2	3	-
CO4	1	1	2	2	3	2	2	2	2	-	3	2	3	1
CO5	1	3	3	2	3	2	2	2	3	2	3	2	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

D. Books and references

1. S. Misra, A. Mukherjee, and A. Roy, 2020. Introduction to IoT. Cambridge University Press. Availability: https://www.amazon.in/Introduction-IoT-Sudip-Misra/dp/1108959741/ref=sr_1_1?dchild=1&keywords=sudip+misra&qid=1627359928&sr=8-1
2. S. Misra, C. Roy, and A. Mukherjee, 2020. Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.
   Availability: https://www.amazon.in/dp/1032146753/ref=sr_1_3?dchild=1&keywords=sudip+misra&qid=1627359971&sr=8-3
3. Research Papers

OCIT3003: BLOCKCHAIN AND ITS APPLICATION (ELECTIVE – 6.1)

Description:

This course OCIT3003: Blockchain And Its Application is offered from SWAYAM as noc25_cs08 – Blockchain and Its Application.

A. Credit and Week:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

B. About the course:

In the last few years, Blockchain technology has generated massive interest among governments, enterprises, and academics, because of its capability of providing a transparent, secured, tamper-proof solution for interconnecting different stakeholders in a trustless setup. In January 2021, the Ministry of Electronics and Information Technology (MeiTY), Government of India, published the first draft of the "National Strategy on Blockchain" that highlights 17 potential applications that are of national interest. Against this backdrop, this subject will cover the basic design principles of Blockchain technology and its applications over different sectors. Additionally, the course also provides tutorials on setting up blockchain applications using one of the well-adopted permissionless blockchain platforms - Ethereum, and one permissioned blockchain platform - Hyperledger.

Pre-requisites:

1. Computer Networks; Operating Systems; Cryptography and Network Security.

C. Course layout:

Week 1: Introduction to Blockchain Technology and its Importance

Week 2: Basic Crypto Primitives I – Cryptographic Hash

Week 3: Basic Crypto Primitives II – Digital Signature

Week 4: Evolution of the Blockchain Technology

Week 5: Elements of a Blockchain

Week 6: Blockchain Consensus I – Permissionless Models

Week 7: Blockchain Consensus II – Permissioned Models

Week 8: Smart Contract Hands On I – Ethereum Smart Contracts (Permissionless Model)

Week 9: Smart Contract Hand On II – Hyperledger Fabric (Permissioned Model)

Week 10: Decentralized Identity Management

Week 11: Blockchain Interoperability

Week 12: Blockchain Applications Monitoring

Course Outcome:

After completion of the course, Students will be able to

CO1	Understand foundational blockchain concepts and basic cryptography behind blockchain technology.
CO2	Understand cryptographic techniques and the historical development of blockchain technology.
CO3	Identify blockchain elements and evaluate consensus algorithms.
CO4	Develop smart contracts and understand decentralized identity management.
CO5	Explore blockchain interoperability and real world applications.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	-	-	1	-	-	2	1	-	-	-	2	-
CO2	-	2	-	-	-	2	1	1	1	-	2	-	1	2
CO3	1	3	2	1	-	2	-	-	2	-	1	2	3	-
CO4	1	1	2	2	3	2	2	2	2	-	3	2	3	1
CO5	1	3	3	2	3	2	2	2	3	2	3	2	3	2

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

D. Books and References

1. Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition, Imran Bashir, Packt Publishing,2020,ISBN:9781839213199,book. Availability:https://www.packtpub.com/product/mastering-blockchain-thirdedition/9781839213199
2. Hyperledger Tutorials - https://www.hyperledger.org/use/tutorials.
3. Ethereum Development Resources - https://ethereum.org/en/developers.
4. Online materials and case studies

IT398: RESEARCH WRITING AND ETHICS (ELECTIVE – 6.2)

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	-	2	-	2	2
Marks	-	100	-	100	

Pre-requisite courses:

Critical Thinking and Analytical Skills, Fundamentals of Academic Writing.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Research Problem Formulation	06
2.	Research Paper Writing	08
3.	Databases and Research Metrics	06
4.	Research Ethics	10
	Total hours (Theory) :	-
	Total hours (Lab) :	30
	Total hours :	30

C. Detailed Syllabus:

1. Research Problem Formulation **06 Hours 25%**

Defining and Formulating the Research Problem: Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem.

2. Research Paper Writing **08 Hours 30%**

Types of research papers, Structure of research papers, Research paper formats, Abstract writing, Methodology, Results and discussions.

3. Databases and Research Metrics **06 Hours 15%**

Definition of Databases, Indexing databases, Citation databases: Web of Science, Scopus, etc. Definition of Research Metrics - Impact Factor of journal as per Journal Citation Report, SNIP, SJR, IPP, Cite Score, Metrics: h-index, g index, i10 index.

4. Research Ethics **10 Hours 30%**

Definition of Plagiarism; Consequences of Plagiarism; Unintentional Plagiarism; Forms of Plagiarism.

D. Student Learning Outcome

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Conduct a thorough literature review, synthesize existing research, and identify gaps in the current knowledge.
CO2	Understand and apply ethical principles in conducting research, including obtaining informed consent, ensuring confidentiality, and avoiding plagiarism.
CO3	Write clear and well-structured research papers and reports, and effectively present their research findings to various audiences.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	1	-	-	-	-	-	-	-	-	-	1	-
CO2	1	2	2	-	-	-	-	-	-	-	-	-	1	-
CO3	1	3	2	-	-	-	-	-	-	-	-	-	1	-

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

E. Recommended Study Material:

- **Text Books:**
  1. American Psychological Association (2010) Publication Manual of the American Psychological Association (6thed.). Washington, DC: APA.
- **Reference Books:**
  1. Day, R. and Castel, B. (2012). How to Write and Publish a Scientific Paper (7thed.). Cambridge: Cambridge University Press.
  2. Research Ethics: A Psychological Approach By Barbara H. Stanley; Joan E. Sieber; Gary B.Melton.

IT399: MASTERING COMPETITIVE PROGRAMMING (ELECTIVE – 6.2)

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	-	2	0	2	**2**
Marks	-	100	0	100	

Pre-requisite courses:

- Data Structures and Algorithms

Objectives of the Course:

- The primary objective of this course is to equip students with a comprehensive understanding of advanced algorithmic techniques essential for competitive programming and placement preparation. By mastering these techniques, students will not only enhance their problem-solving skills but also gain a competitive edge in coding competitions such as the ICPC.

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Range Queries	04
2.	Algorithms on tree data structure	05
3.	Algorithms on graph data structure – 1	04
4.	Algorithms on graph data structure – 2	04
5.	Greedy programming	04
6.	Dynamic programming	06
7.	Additional topics	03
	Total hours (Theory):	00
	Total hours (Lab) :	30
	Total hours :	30

C. Detailed Syllabus:

1. Range Queries **04 Hours** **12%**

Queries on static arrays (sum and minimum queries), tree structures (binary indexed trees, segment trees), additional techniques

2. Algorithms on tree data structure **05 Hours** **15%**

Basic techniques (traversal, diameter, all longest paths), tree queries (find ancestors, subtrees, lowest common ancestors)

3.	**Algorithms on graph data structure – 1** Basic of graph (graph terminologies, representation), graph traversal (DFS, BFS, applications), revision of shortest paths, successor graphs, MST,	**04 Hours**	**12%**
4.	**Algorithms on graph data structure – 2** Traversal problems (finding connected components, flood fill, bipartite graph check, articulation points and bridges, strongly connected components), single source shortest paths, all pair shortest paths, special graphs (DAG, eulerian, bipartite)	**04 Hours**	**20%**
5.	**Greedy programming techniques** Coin change revisited, load balancing, interval covering, sort the input first	**04 Hours**	**12%**
6.	**Dynamic programming** Finding an optimal solution, longest increasing subsequence, paths in grid, permutations to subsets, counting tilings, classical examples	**06 Hours**	**20%**
7.	**Additional topics** Square root techniques, segment trees, dynamic programming optimization, miscellaneous	**03 Hours**	**9%**

D. Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Solve and apply concept of tree and graph data structure for solving real time problems
CO2	Apply greedy algorithms to solve optimization problem
CO3	Apply dynamic programming techniques for solving complex problems.
CO4	Design efficient algorithms and develop strong problem-solving skills.
CO5	Compete in competitive programming contests and coding interviews.

Course Articulation Matrix:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	1	-	1	-	-	-	-	-	-	-	-	2	1	1
CO2	2	2	-	-	-	-	-	-	-	-	-	2	2	-
CO3	3	3	3	3	2	-	-	-	-	-	-	2	2	-
CO4	2	2	-	-	-	-	-	-	-	-	-	-	3	-
CO5	3	3	-	1	-	-	-	-	-	-	-	-	2	-

The correlation levels 1, 2, or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, "-" is placed.

E. Recommended Study Material:

- **Text book:**
    1. Competitive programming by Steven & Felix Halim.
    2. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests.
- **Web Link:**
    1. www.ieeexplore.ieee.org
    2. www.sciencedirect.com

HS132.02 A: CONTRIBUTORY PERSONALITY DEVELOPMENT

A. Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	--	02	--	30	**02**
Marks	**--**	100	--	100	

Pre-requisite courses:

- Communication and Soft Skills

Objectives of the Course:

- Become familiar with basic concept of personality and personality development
- Understand personality development theories and strategies
- Evaluate one's personality and inculcate traits of an assertive personality
- Develop an assertive personality
- Develop life skills and required management traits
- Enhance contributory personality for academic and career success

B. Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Concept of Personality	06
2.	Soft Skills and Personality Development	08
3.	Developing Contributory Personality	06
4.	Life skills and Personality Development	06
5.	Contemporary Issues in CPD	04
	Total hours (Theory) :	**--**
	Total hours (Practical) :	**30**
	Total hours :	**30**

C. Detailed Syllabus:

1.	**Concept of Personality** Meaning of Personality, Types of Personality, Factors contributing to Personality, Personality Traits, Personality Profiling	06 Hours	20%
2.	**Soft Skills and Personality Development** Positive Thinking and Mind Set, Leadership, Assertiveness and Negotiation Skills, Self-Management, Interpersonal Skills, Being a Team Player	08 Hours	26%

3.	**Developing Contributory Personality** Concept of Contributory Personality, Characteristics of a Contributor, The Contributor's Vision of Success & Career, The Scope of Contribution in a field, Embarking on the Journey to Contributor ship, Developing Contributor Personality, Reviewing Some Contributors Personalities	06 Hours	20%
4.	**Life skills and Personality Development** Concept of life skills, Self-awareness, Empathy, Decision Making, Problem Solving	06 Hours	20%
5.	**Contemporary Issues in CPD** Contemporary Trends and Practices in Contributory Personality Development, Case Study & Presentations	04 Hours	14%

D. Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Identify one's individual personality strengths and challenges.
CO2	Develop more assertive and optimist attitude towards work and life.
CO3	Develop quintessential soft skills to groom one's personality.
CO4	Identify traits of contributor personality.
CO5	Contribute to self, society, nation, and globe.
CO6	Develop skills of global citizenship to perform societal responsibilities.

E. Recommended Study Material:

❖ **Textbook:**

1. Personality Development & Soft Skills, Oxford University Press
2. Soft Skills, Bookboon
3. Personality Development, Swami Vivekananda; Advaita Ashrama

❖ **Reference book:**

1. Contributor Personality Program Workbook (Volume 1,2),
2. Contributor Personality Program ActivGuide, Illumine Knowledge Pvt. Ltd

❖ **Web material:**

1. https://www.coursera.org/learn/wharton-succcess
2. https://www.coursera.org/learn/personality-types-at-work
3. https://www.coursera.org/learn/self-awareness

