Fourth Semester					
Theory					
Sl. No.	**Category**	**Course Code**	**Course Title**	**L-T-P**	**Credit**
1	IT	19MC401	Design and Analysis of Algorithms	3-0-0	3
2	IT	19MC402	Software Engineering	3-0-0	3
3	IT	19MC403	Introduction to Machine Learning	3-0-0	3
4	IT	19MC404	Elective-I	3-0-0	3
Total Credit (Theory)					12

Practical					
1	LAB	19MC405	Design Analysis of Algorithms Lab	0-0-4	2
2	LAB	19MC406	Software Engineering Lab	0-0-4	2
3	BM	19MC407	Soft Skill Development - II	0-0-4	2
4	SEM	19MC408	Seminar and Technical Writing	0-0-4	2
Total Credit (Practical)					8
Total Semester Credit					20

Fourth Semester
Theory

Course Code: 19MC401	Course Name: Design and Analysis of Algorithm	L-T-P 3-0-0	Credit 3

Course Objective:

1. Translating a plain text problems to convert into an algorithm
2. Calculate best case, worst case time complexity and space complexities of different algorithm and choosing the best solution from the available options
3. Applying different design paradigm to solve different problems and comparing their best case, worst case scenarios.
4. Designing and applying different data structures over different algorithms for solving different problems.
5. Understand different P-class, NP class problems.

Module-1: **(12 Hrs.)**

Introduction to problems and algorithms , Mathematics for algorithm analysis , Insertion sort Analysing algorithms, Designing of algorithms, Asymptotic notation Standard notations and common functions, Recurrence relations, The substitution method, The recursion-tree method, The master method, Divide and conquer: Min-Max Heap, Priority queue,Heapsort , Quicksort, Merge Sort, Sorting in Linear Time: Lower bounds for sorting: Counting sort, Radix sort, Bucket sort, Fast Fourier transform , Finding the convex hull : Graham Scan, Finding the closest pair of points

Module-2: **(8 Hrs.)**

Greedy method: Elements of the greedy strategy, Huffman codes, task-scheduling problem, Fractional Knapsack problem, Coin change problem, Dynamic programming: Assembly-line Scheduling, Matrix-Chain Multiplication, Longest Common Subsequence(LCS), 0/1 Knapsack problem, Rod Cutting problem

Module-3: **(6 Hrs.)**

Graph algorithms: Basic Definitions and Application, Representations of graphs, Breadth-first search and Depth-first search, Data Structures for Disjoint Sets, Strongly connected components, Minimum Spanning Trees: The algorithms of Kruskal and Prim

Module-4: **(6 Hrs.)**

Single-Source Shortest Paths: The Bellman-Ford algorithm, Dijkstra's algorithm, All-Pairs Shortest Paths-Shortest paths and matrix multiplication, The Floyd-Warshall algorithm

String Matching: The naive string-matching algorithm, The Rabin-Karp algorithm, The Knuth-Morris-Pratt algorithm

Module-5: **(8 Hrs.)**

Network Flow: Flow networks, The Ford-Fulkerson method, Maximum bipartite matching
Backtracking – n-Queen problem – Hamiltonian Circuit Problem – Subset Sum Problem.
Branch and Bound – LIFO Search and FIFO search – Assignment problem – Knapsack Problem, NP-Completeness: Classes P and NP, NP-complete problems.: Reduction of 3SAT to Subset Sum, Approximation Algorithm for TSP

Course Outcome:

1. Given a English language problem description define the problem precisely with input/output requirements, examine its inherent complexity and develop a generic or set of initial solutions and justify their correctness.
2. Given an algorithm descriptions, analyse the time and space complexity of the algorithm in the worst case, average case, and amortized scenario as needed in terms of asymptotic order of complexity.
3. Given a problem definition explore different alternative algorithmic solutions, compare them with respect to time and space complexity and choose the design scheme and /or design parameter and data structure appropriately to obtain the best possible choice(s) that can be converted to an executable programs.
4. Examine and prove whether a problem is of polynomial complexity, hard(np complete) or otherwise and develop optimal and approximate algorithm for them as applicable.

Suggested Books:

1. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, MIT Press/McGraw-Hill, 2009.
2. Ellis Horowitz, SartajSahni and SanguthevarRajasekaran, Computer Algorithms/ C++, Second Edition, Universities Press, 2007.
3. SanjoyDasgupta, Christos H. Papadimitriou and Umesh V. Vazirani, Algorithms, McGraw-Hill, 2008.
4. Jon Kleinberg and ÉvaTardos, Algorithm Design, Addison-Wesley/PEARSON EDUCATION-2006.
5. S. Sridhar, —Design and Analysis of Algorithms‖, Oxford university press, First Edition, 2015.

MOOC:

1. Prof.Abhiram G Ranade, Prof.Ajit A Diwan, Prof.SundarViswanathan,IIT Bombay, https://nptel.ac.in/courses/106101060/

2. Prof.MadhavanMukund,Chennai Mathematical Institute,https://nptel.ac.in/courses/106106131/

3. Reyna Hulett, CS161, Stanford School of Engineering, https://online.stanford.edu/courses/cs161-design-and-analysis-algorithms

Course Code: 19MC402	Course Name: Software Engineering	L-T-P 3-0-0	Credit 3

Course Objective:

1. Explain different software development paradigm
2. Demonstration of UML diagrams and its pros and cons
3. Explain different metrics' used in project management
4. Explain different software engineering practices.

Syllabus

Module-1: (8 Hrs.)

Introduction, Introduction to Software Development processes, Agile software development: Agile methods, Plan-driven and agile development, Extreme programming, Agile Process model: Adoptive software development, scrum, crystal, Agile modelling, Agile unified process.

Module-2: (6 Hrs.)

Requirements engineering: Functional and non-functional requirements: The software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management

Module-3: (10 Hrs.)

Object-oriented design using UML: Analysis and Design: Concepts, Classes and Objects. Relationships Among Objects. Inheritance and Polymorphism, Design Concepts, Design Notation and Specification, Design Methodology, Dynamic Modelling, Functional Modelling, Defining Internal Classes and Operations, Design patterns.
System modelling: Context models, Interaction models, Structural models, Behavioural models Model-driven engineering

Module-4: (6 Hrs.)

Architectural design: Architectural design decisions, Architectural views, Architectural patterns, Application architectures, Design and implementation,
Testing: Introduction to software testing, verification and validation, unit testing, integration testing, system testing. Software Maintenance.

Module-5: (10 Hrs.)

Project management: introduction to Risk management, managing people, Teamwork, Project planning, Software pricing, Plan-driven development, Project scheduling, Estimation techniques, Quality management, Software measurement and metrics

Introduction to Advanced Software Engineering concepts: Software reuse, Component-based software engineering, Distributed software engineering, Service-oriented architecture, Embedded software, Aspect-oriented software engineering

(no such restriction of 4 modules)

Course Outcome:

1. Design UML diagram for given problem scenario
2. Translating a plan text problem to functional requirements.
3. Evaluation of software models for specific problems

Suggested Books:

1. Ian Sommerville *Software Engineering-,* Pearson Education, New Delhi, 10th Edition 2017, ISBN: 9789332582699
2. Roger S. Pressman, Bruce R. Maxim,*Software Engineering-A practitioner's approach-* McGraw-Hill International Editions, New York. ISBN: 9789353165710, 9353165717, Edition: 8, 2019
3. Pankaj Jalote , *An Integrated Approach to Software Engineering-*, Narosa Publishing House. Third Edition, 978-81-7319-702-4 , Reprint 2019
4. RajibMall ,*Fundamentals of Software Engineering-*, PHI, New Delhi. 5th Edition, 2018
5. UgrasenSuman *Software Engineering: Concepts & Practices, Cengage Learning publications. 1st Edition, 2013.*

MOOC:

1. Prof.Rajib Mall, IIT Kharagpur, https://nptel.ac.in/courses/106105182/, https://nptel.ac.in/syllabus/106105087/, https://swayam.gov.in/courses/4736-july-2018-software-engineering
2. Prof.Rushikesh K Joshi, Prof.UmeshBellur,Prof. N.L. Sarda,IIT Bombay, https://nptel.ac.in/syllabus/106101061/
3. Praveen Mittal, Kevin Wendt, College of Science and Engineering,Department of Computer Science and Engineering, University of Minnesota https://www.coursera.org/learn/software-processes
4. Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch, Instructor TechnischeUniversitätMünchen , https://www.edx.org/course/software-engineering-essentials

Course Code: 19MC403	Course Name: Machine Learning	L-T-P: 3-0-1	Credit: 3

Course Objective:

1. Explain Supervised learning and unsupervised learning models
2. Explain neural network-based classification and regression problems
3. Demonstration of models using R / Python / Matlab programs
4. Explaining Feature generation and feature evaluation

Syllabus

Module I: **[12 Hours]**

Introduction and Supervised Learning: Concepts of Machine Learning, Example of Machine Learning Applications like Associations, Classification, Regression, Unsupervised Learning, and Reinforcement Learning.

Supervised Learning- Artificial Neural Network: Biological Motivation, ANN Representation, Appropriate Problem for ANN Learning, Perceptron, Multilayer Networks and the Backpropagation Algorithm, Remarks on the Back-propagation Algorithm, Classifying with k-Nearest Neighbour Classifier, Support Vector Machine Classifier, Decision Tree Classifier, Naive Bayes Classifier, Bagging, Boosting.

Module II: **[8 Hours]**

Forecasting and Learning Theory: Predicting Numeric Values: Regression, Linear Regression, Logistic-Regression, Tree-Based Regression. Bias/Variance Trade-off, Union and Chernoff/Hoeffding Bounds, Vapnik–Chervonenkis(VC) Dimension, Support Vector Machines for Regression.

Module-III: **[6 Hours]**

Unsupervised Learning: Grouping Unlabelled Items using k-Means, Adaptive Hierarchical Clustering, Density Based Clustering, Cluster Validation (Inter-Intra Cluster Distance, Dunn Index), Choosing the Number of Clusters.

Module-IV: **[8 Hours]**

Dimensionality Reduction: Dimensionality Reduction - Feature Extraction, Factor Analysis - Principal Component Analysis, Singular Value Decomposition. Feature Selection – Feature Ranking and Subset Selection, Filter, Wrapper and Embedded Methods.

Module-V: **[6 Hours]**

Reinforcement Learning: Introduction, Elements of Reinforcement Learning, Model-Based Learning (Value Iteration and Policy Iteration), Nondeterministic Rewards and Actions, Temporal Difference Learning, Relationship to Dynamic Programming.

Course Outcome:

1. To explain supervised (classification, regression, prediction) and unsupervised (clustering, feature reduction) problems
2. Implement classification, regression and clustering algorithms.
3. Performance analysis of various classification, regression and clustering algorithms.

Suggested Books:

1. Introduction to Machine Learning Second Edition, Ethem Alpaydin, PHI, 2010.
2. Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, MIT Press, 2012.
3. Pattern Recognition and Machine Learning, Christopher Bishop, Springer, 2007.
4. Neural Networks and Learning Machines, Simon Haykin, PHI, 2010.
5. Machine Learning First Edition, Tom Mitchell. McGraw- Hill, 1997.
6. Machine Learning in Action, Peter Harrington, Manning Publication, 2012.

7. https://onlinecourses.nptel.ac.in/noc18_cs26/preview
   (Prof. BalaramanRavindran, Indian Institute of Technology Madras)
8. https://onlinecourses.nptel.ac.in/noc18_cs40/preview
9. (Prof. Sudeshna Sarkar, Indian Institute of Technology Kharagpur.

Elective – I

Course Code: 19MC404A	Course Name: Software Testing	L-T-P: 3-0-0	Credit: 3

Course Objective:

1. To learn the evolution of software testing techniques, myths and facts of software testing, models for testing processes, various types of software testing.
2. To design test cases using black-box and white-box testing techniques
3. To understand basic concepts of regression testing, problems of regression testing and types of regression testing techniques.
4. To learn the strategies for testing of object-oriented applications and web-based applications
5. To learn about test automation and present day practices

Syllabus

Module-1: Introduction (10 Hrs.)

Testing as an Engineering Activity, Testing Fundamentals, Defects, Strategies and Methods for Black Box Test Case Design, Strategies and Methods for White-Box Test Case design, Test Adequacy Criteria, Evaluating Test Adequacy Criteria, Levels of Testing and different types of testing, OO Testing.

Module-2: Test Case Design (11 Hrs.)

The need for levels of testing, Testing Lifecycle, Unit Test, Unit Test Planning, Designing the Unit Tests, The test harness, Running the Unit tests and Recording results, Integration tests, Designing Integration Tests, Integration Test Planning, Scenario testing, Defect bash elimination System Testing, Acceptance testing, Performance testing, Regression Testing, Internationalization testing, Ad-hoc testing,

Alpha, Beta Tests, Testing OO systems, Usability and Accessibility testing, Configuration testing, Compatibility testing, Testing the documentation, Website testing

Module-3: Test Management (8 Hrs.)

Testing and Debugging Goals and Policies, Test Planning, Test Plan Components, Test Plan Attachments, Locating Test Items, Reporting Test Results, Process and the Engineering Disciplines, Introducing the test specialist, Skills needed by a test specialist, Building a Testing Group, Measurement and Milestones for Controlling and Monitoring, Status Meetings, Reports and Control Issues, Criteria for Test Completion, SCM, Types of reviews, Developing a review program, Components of Review Plans, Reporting review results.

Module-4: Software Quality (4 Hrs.)

Software quality definition, Software quality assurance, Software quality factors, Components of the SQA system, SQA standards, System certification and assessment components, Considerations guiding construction of an organization's SQA system.

Module-5: Test Automation (7 Hrs.)

Software test automation, Skills needed for automation, Scope of automation, API testing, CLI testing, GUI testing, System Under Test, Test automation architecture, Test automation framework, Test automation strategy, Test script, Requirements for a test tool, Challenges in automation, Test metrics and measurements, project, progress and productivity metrics.

Test automation using Testing tool (Hands-on session on Selenium) – IDE, WebDriver, TestNG, Introduction to some other popular test automation tools like Telerik Test Studio, Robotium, TestComplete, Watir, Visual Studio Test Professional, QTP (UFT), SoapUI, FitNesse, TestDrive,

Note: Questions from **Test automation using Testing tool** may not come for exams.

Course Outcome:

After reading this subject, students will be able to:

1. Understand the evolution of software testing techniques, their goals and learn the various models of software testing.
2. Generate test cases for software systems using black box and white box testing techniques.
3. Carry out regression testing of software systems.
4. Test conventional, object-oriented and web based software.
5. Understand debugging software and types of debuggers
6. Understand how software test automation is done

Suggested Books:

1. Srinivasan Desikan and Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson Education, 2006.
2. Naresh Chauhan, Software Testing Principle and Practices, Oxford University Press, 2016.
3. Ron Patton, “Software Testing”, Second Edition, Sams Publishing, Pearson Education,
   2007.
4. Ilene Burnstein, “Practical Software Testing”, Springer International Edition, 2003.
5. Martin Wieczorek, & Dirk Meyerhoff, “Software Quality”, Springer, 2001.
6. Elfriede Dustin, Jeff Rashka, & John Paul, “Automated Software Testing”, Addison Wesley, 1999.
7. Rahul Shende, “Software Automation Testing Tools for Beginners”, SPD, 2012.

MOOC course:
1. https://www.edx.org/course/software-testing-fundamentals-usmx-umuc-stv1-1x-4
   by-
Dr. Michael Scott Brown, Program Chair of the Software Engineering Master’s University of Maryland University College
Dr. Renata Rand McFadden, Adjunct Professor University of Maryland University College

Course Code: 19MC404B	**Course Name: Python Programming**	**L-T-P: 3-0-0**	**Credit: 3**

Course Objective:

1. To acquire programming skills in core Python.
2. To acquire Object Oriented Skills in Python
3. To develop the skill of designing Graphical user Interfaces in Python
4. To develop the ability to write database applications in Python

Module-I

Python: Features of Python , Installing Python for windows and setting up paths, writing and Executing of a python programs, Python Virtual machine, Frozen binaries, Comparison between C, Java and python , Comments , Docstrings ,How python sees variables, Data types in Python, built in types, sequences in python, sets, literals in Python, user defined data types, identifiers & reserved words, Naming convention in python,

Module-II

various Operators in Python, Input & Output, Control statements, if statements, while loop, for loop, infinite loop, nested loop ,else suit, break, continue, pass ,assert, return statements, command line arguments.

Arrays in python, advantages using arrays, creating arrays, importing the array module, indexing and slicing on arrays, Processing the arrays, Comparing arrays.

Strings in Python, Creating strings, Length of a string, Indexing in strings, Slicing strings, Concatenation and Comparing strings, Finding Substrings, Replacing a String.

Module-III

Functions in Python, define a function, calling a function, return from function, pass by object Reference, Positional arguments, Default arguments, Recursive functions.

Introduction to OOP, features of OOP, creating classes, the self-variable, constructor, types of variables, namespaces, types of methods.

Module-IV

Inheritance: Define inheritance, types of inheritance, constructors in inheritance, overriding super class constructors & methods, the super() method, MRO

Polymorphism: Duck typing philosophy of Python, operator overloading, method overriding, interfaces in python

Exceptions: Errors in a python program, Exceptions, Exception handling, Types of Exceptions, The Exception block, the assert statement, user defined exceptions

Python Database Connectivity: DBMS, types of databases used with Python, installation of MySQL database, setting path, verifying MySQL , installing MySQL connector, Working with MySQL database, Using MySQL from python, retrieving rows ,deleting rows, updating rows in a table.

Text Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011.
2. Core Python Programming, Dr. R. Nageswar Rao , Dreamtech Press
3. Python Programming for Absolute Beginners, Michael Dawson, CENGAGE Learning

Reference Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012

Course Name	Python Programming
Course Material Link	Python Tutorial/Documentation www.python.or 2015
Course Material Link	http://docs.python.org/3/tutorial/index.html

Course Material Link	http://interactivepython.org/courselib/static/pythonds

Practical

Course Code: 19MC405	Course Name: Design and Analysis of Algorithm Lab	L-T-P 0-0-2	Credit: 2

1. Insertion Sort/ Selection Sort
2. Divide and Conquer: Fibonacci search/Binary search
3. Divide and Conquer: Merge Sort/Quicksort/Heap Sort
4. Divide and Conquer: Convex hull/Finding closet pair
5. Dynamic Programming: MCM/LCS
6. Dynamic Programming: Rod Cutting problem /Assembly line Scheduling
7. Greedy method: Activity Selection/Huffman Coding
8. Graph Search: BFS/DFS
9. Graph Greedy MST: Kruskal/Prim's
10. Graph Greedy Shortest Path: Bellman ford/Dijkstra
11. Rabin Karp string matching algorithm/Subset Sum problem using Branch and Bound

Prerequisite: Each student should have a good knowledge on basic data structures like Stack, Queue, List, Heap, Matrix

Course Code: 19MC406	Course Name: Software Engineering Lab	L-T-P: 0-0-2	Credit: 2

1. **Identifying the Requirements from Problem Statements**

Requirements | Characteristics of Requirements | Categorization of Requirements | Functional Requirements | Identifying Functional Requirements

2. **Estimation of Project Metrics**

Project Estimation Techniques | COCOMO | Basic COCOMO Model | Intermediate COCOMO Model | Complete COCOMO Model | Advantages of COCOMO | Drawbacks of COCOMO | Halstead's Complexity Metrics

3. **Modelling UML Use Case Diagrams and Capturing Use Case Scenarios**

Use case diagrams | Actor | Use Case | Subject | Graphical Representation | Association between Actors and Use Cases | Use Case Relationships | Include Relationship | Extend Relationship | Generalization Relationship | Identifying Actors | Identifying Use cases | Guidelines for drawing Use Case diagrams

4. **E-R Modelling from the Problem Statements**

Entity Relationship Model | Entity Set and Relationship Set | Attributes of Entity | Keys | Weak Entity | Entity Generalization and Specialization | Mapping Cardinalities | ER Diagram | Graphical Notations for ER Diagram | Importance of ER modelling

5. **Identifying Domain Classes from the Problem Statements**

Domain Class | Traditional Techniques for Identification of Classes | Grammatical Approach Using Nouns | Advantages | Disadvantages | Using Generalization | Using Subclasses | Steps to Identify Domain Classes from Problem Statement | Advanced Concepts

6. **Statechart and Activity Modelling**

Statechart Diagrams | Building Blocks of a State chart Diagram | State | Transition | Action | Guidelines for drawing State chart Diagrams | Activity Diagrams | Components of an Activity Diagram | Activity | Flow | Decision | Merge | Fork | Join | Note | Partition | A Simple Example | Guidelines for drawing an Activity Diagram

7. **Modelling UML Class Diagrams and Sequence diagrams**

Structural and Behavioral aspects | Class diagram | Elements in class diagram | Class | Relationships | Sequence diagram | Elements in sequence diagram | Object | Life-line bar | Messages

8. **Modelling Data Flow Diagrams**

Data Flow Diagram | Graphical notations for Data Flow Diagram | Explanation of Symbols used in DFD | Context diagram and leveling DFD

9. **Estimation of Test Coverage Metrics and Structural Complexity**

Control Flow Graph | Terminologies | McCabe's Cyclomatic Complexity | Computing Cyclomatic Complexity | Optimum Value of Cyclomatic Complexity | Merits | Demerits

10. **Designing Test Suites**

Software Testing | Standards for Software Test Documentation | Testing Frameworks | Need for Software Testing | Test Cases and Test Suite | Types of Software Testing | Unit Testing | Integration Testing | System Testing | Example | Some Remarks

Course Code: 19MC407	**Course Name:** Soft Skills Development-II	**L-T-P:** 0-0-2	**Credit:** 2

Course Objectives:

1. To understand the importance of values and ethics in organisational communication.
2. To be acquainted with the general preparation required for interviews.
3. To master the art of public speaking.

4. To understand the procedure required for conducting a professional meeting.
5. To increase learners' soft skills so as to develop their attributes for professional communication.

Syllabus Details:

1. **Professional Ethics and Values**: What are ethics and values? Fairness: to behave in an open, just and respectable way toward other people, Oneness and respect for individual, Helpfulness, Social Responsibility, Inclusiveness/ Belongingness.
2. **Interpersonal Skills and Behavioural Skills**: Importance of inter-personal skills, Positive relationship, Positive attitudes, empathise: Comprehend other's points of view and face them with understanding, Mutuality, Trust, Emotional bonding, Handling situations.
3. **Professional Life Skills_1:** *Stress Management*- its causes, Different kinds of stress, Techniques to cope with the stressful situation at a workplace. *Self-Management*: Self Evaluation, Self-discipline, Self-criticism, Self-awareness, Recognition of one's own limits and deficiencies, Managing self: emotions, ego, pride etc.
4. **Professional life Skills_2**: *Time Management*- Goal setting, how to prepare the time line and allocate time to complete different tasks, how to successfully follow the prepared time-schedule, *Multi-Tasking:* How to prioritize the work, Importance of multi-tasking and concerns related to multi-tasking, Identifying what to multi-task,

5. **Interview Skills:** Preparing a resume (functional and combination), preparing Video Resume, FAQs of Interview and ways to handle them, Teleconferencing and Video Conferencing Skills, Mock interview.
6. **Organising Meetings**: Planning a meeting, organising a meeting, Process of a meeting, designing agenda and preparing minutes of the meeting. Strategic issues related to effective meeting.
7. **Public Speaking**: Art of public speaking, Rhetoric of making a public speech, Exploring rhetorical elements through various videos. Controlling nervousness and stage fright.

Course Outcomes:

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

1. Implement professional ethics and values in communication.
2. Face the job interviews confidently.
3. Address the audience confidently and effectively.
4. Prepare agenda, conduct meeting and draft minutes of the meeting.
5. Inculcate potential to deal with the external world in a collaborative manner creating a positive impact and good impression.

Suggested Reading:

1. Alex, K. *Soft skills*. S. Chand Publishing, 2009.
2. Bhatnagar, Nitin, ed. *Effective communication and soft skills*. Pearson Education India, 2011.
3. Ghosh, B. N. "Managing Soft Skills for Personality Development." *Tata McGraw.*
4. Joshi, Manmohan. "Soft skills." 2017.
5. McLean, S., and M. Moman. "Communication for Business Success (Canadian Edition)." 2012.
6. Ramesh, Gopalaswamy. *The ace of soft skills: attitude, communication and etiquette for success*. Pearson Education India, 2010.
7. Rao, Manchanahalli Satyaranayana. *Soft skills-enhancing employability: connecting campus with corporate*. IK International Pvt Ltd, 2010.
8. Riggio, Ronald E., and Sherylle J. Tan, eds. *Leader interpersonal and influence skills: The soft skills of leadership*. Routledge, 2013.
9. Sharma, Prashant. *Soft Skills: Personality Development for Life Success*. BPB publications, 2019.
10. Vikram Bisen, Priya. "887. Business Communication." (2009).

Course Code: 19MC408	Course Name: Seminar and Technical Writing	L-T-P: 0-0-2	Credit: 2

Introduction to Technical Writing: Definition, History of Technical Writing, Characteristics of Technical Writing, Importance of Technical Writing, Types of Technical Documents, Difference between Technical and Non-technical Writing.

Language in Technical Writing: Analysis and Mechanism of Elements in Sentences-the Parts of Sentences, the Clauses, the Correct Verb Usage: Tense, Voice and Mood, Coordination and Subordination, Pronoun Reference and Antecedents, Recursion, Nominalization.

Different Kinds of Technical Writing: research papers, user manuals, thesis, various Technical Reports, Technical E-mails, Project Proposals, Brochures, Questionnaires, PowerPoint and poster presentation. Brainstorming, Note-Taking, Writing Paragraphs: Structure and Development, Methods of Development, the Paraphrase of Literature Review and its mechanics (et.al, ibid…,), Plagiarism.

Style and Structure of Technical Writing: Style and Format, A Review of Common Errors (ESL trouble spots) and Tips for style and word choice, Terminology and Jargon, Bibliographies and References, the Mechanics: Punctuations, Acronym and Abbreviation, Figures (Graphs) and Tables, Flowcharts, Sections and Section Numbering, and other elements in technical writing, Assessing Audience and Context, Proofreading, Checklist for Proofreading.

C. Text Books:

1. Gerald J. Alred, et. al, Handbook of Technical Writing, 10th Edition, Publisher: IEEE, 2011.

2. Phillip A. Laplante, Technical Writing, A Practical Guide for Engineers and Scientists, Taylor & Francis, 1st Edition, 2012.

D. Reference:

1. Biber, D., & Gray, B.. Grammatical complexity in academic English: Linguistic change in writing, Cambridge: Cambridge University Press. 2016.

2. Ken Hyland, Feng (Kevin) Jiang, Is academic writing becoming more informal? English for Specific Purposes 45 40–51. 2017.

3. Bennet, K,. English academic style manuals: A survey, Journal of English for Academic Purposes, 8, 43-54. 2009.

4. S. Ravilla. The Elements of International English Style: A Guide to Writing English Correspondence, Reports, Technical Documents, and Internet Pages for a Global Audience, IEEE