



CIRCULAR NO.SU/Sci. & Tech./Colleges./NEP/16/2023

It is hereby inform to all concerned that, the syllabi prepared by the Board of Studies & Ad-hoc Boards and recommended by the Dean, Faculty of Science & Technology, the Hon'ble Vice-Chancellor has accepted **the following curriculum of All Post Graduate Degree Courses as per Norms of National Education Policy - 2020 under the Faculty of Science & Technology run to the Affiliated Colleges, Dr.Babasaheb Ambedkar Marathwada University** in his emergency powers under section 12(7) of the Maharashtra Public Universities Act, 2016 on behalf of the Academic Council as appended herewith.

Sr.No.	Syllabi of Affiliated BAMU, Aurangabad.	Semester
1.	M.Sc.Zoology	Ist and IInd Semester
2.	M.Sc.Microbiology	Ist and IInd Semester
3.	M.A/M.Sc.Mathematics	Ist and IInd Semester
4.	M.Sc.Geology	Ist and IInd Semester
5.	M.Sc.Biophysics	Ist and IInd Semester
6.	M.Sc.Bioinformatics	Ist and IInd Semester
7.	M.Sc. Information Technology	Ist to IVth Semester
8.	M.Sc.Computer Science	Ist to IVth Semester
9.	M.Sc.Botany	Ist and IInd Semester
10.	M.Sc.Environmental Science	Ist and IInd Semester

This is effective from the Academic Year 2023-24 and onwards.

All concerned are requested to note the contents of this circular and bring the notice to the students, teachers and staff for their information and necessary action.

University Campus,
Aurangabad-431 004.

REF.NO.SU/NEP/2023/ 8752-60

Date:- 08.08.2023.

★
★
★
★

*Deputy Registrar,
Academic Section*

Copy forwarded with compliments to :-

- 1] The Principal of all concerned affiliated Colleges, Dr. Babasaheb Ambedkar Marathwada University,.
- 2] The Director, University Network & Information Centre, UNIC, with a request to upload this Circular on University Website.

Copy to :-

- 1] The Director, Board of Examinations & Evaluation, Dr.BAMU,A'bad.
- 2] The Section Officer,[M.Sc.Unit] Examination Branch,Dr.BAMU,A'bad.
- 3] The Programmer [Computer Unit-1] Examinations, Dr.BAMU,A'bad.
- 4] The Programmer [Computer Unit-2] Examinations, Dr.BAMU,A'bad.
- 5] The In-charge,[E-Suvidha Kendra], Rajarshi Shahu Maharaj Pariksha Bhavan, Dr.BAMU,A'bad.
- 6] The Public Relation Officer, Dr.BAMU,A'bad.
- 7] The Record Keeper, Dr.BAMU,A'bad.

**DR. BABASAHEB AMBEDKAR MARATHWADA
UNIVERSITY, AURANGABAD**



NAAC Reaccredited A Grade

FACULTY OF SCIENCE & TECHNOLOGY

**2 Years / 1 Year M.Sc. Information Technology
Course Structure (For University Affiliated Colleges)**


Dean
Faculty of Science & Technology
Dr. Babasaheb Ambedkar Marathwada
University, Aurangabad



(Effective from 2023-24)

COURSE STRUCTURE AS PER GUIDELINES OF NEP 2020

Illustrative Credit distribution structure for two/ one year **M.Sc. Information Technology** Programme with Multiple Entry and Exit options for Discipline Specific Course for Colleges.

- **Eligibility for the course:**

- i) B.Sc. Computer Science (OR) B.Sc. Information Technology (OR) B. Sc. Computer Application (Science & Technology) (OR) B.E/B. Tech. in Computer Science and Engineering/IT. (OR) ii) Any Science Graduate with at least one Optional Subject as Computer Science or Any Science Graduate having Mathematics as one of the subject at HSC(XII)

- **Credits and Degrees:**

- i) A candidate who has successfully completed all the core courses, Elective / Specialized courses and, seminars and project prescribed and or optional service courses approved by the University for the program with prescribed CGPA shall be eligible to receive the degree.
- ii) One Credit shall mean one teaching period of one hour per week for one semester (of 15 weeks) for theory courses and two practical / laboratory / field / demonstration hours / week for one semester.
- iii) Every student will have to complete 88 credits to obtain the master's degree of the said programme.
- iv) The college is committed towards ensuring the provision of the necessary contact hours for course engagement by the faculty. Accordingly, the workload for both the course and faculty will be calculated, taking into account the allocated contact hours for one credit in theory as well as practical components as per following
 - a. 1 Credit (THEORY) = 15 Contact Hours / Semester i.e 1 Contact hour per week
 - b. 1 Credit (PRACTICAL) = 30 Contact Hours / Semester, i.e 2 Contact hour per week per batch. The batch size for practical will be min 8 student – maximum 10 students

However, the course is framed the curriculum as per the Model course structure suggested by NEP 2020 guidelines.

- **Courses Inclusions:**

- (i) **Core Course / Mandatory Courses / Discipline Specific Core Courses (DSC):** - A core course is a course that a student admitted to M. Sc. Computer Science/ M. Sc. Information Technology program must successfully completed to receive the degree. Normally no theory course shall have more than 4 credits.
- (ii) **Discipline Specific Elective Course (DSE):** Means Elective course from the basic subject or specialization. The elective course defined for 4 credits and dedicated for choice of specialization that student want to perceive. The horizontal learning path is to be followed by the student for selection of elective course. Department may offer more than one specialization depending availability of resources.
- (iii) **Skill Courses (SC):** The skill courses will be offered as per the structure of the NEP. This course will be conducted to impart special technology skills to the students. This course is defined for 2 Credits and completely engaged in practical form.
- (iv) **On Job Training (OJT):** The student is required to complete 30 hours on job training in the summer of semester 2 examination.
- (iv) Each Course shall include lectures / tutorials / laboratory or field work / Seminar / Practical training / Assignments / midterm and term end examinations/ paper / Report writing or review of literature and any other innovative practice etc., to meet effective teaching and learning needs.

PROGRAM OBJECTIVES (PO) for M.Sc. Information Technology

Program objectives for an M.Sc. (Master of Information Technology) in Computer Science course typically aim to provide students with a comprehensive understanding of computer science concepts and practical skills. Following are some broad program objectives earmarked by the department:

1. Apply the knowledge of mathematics, science and computing in the core information technologies.
2. Identify, design, and analyze complex computer systems and implement and interpret the results from those systems.
3. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
4. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
5. Communicate effectively in a variety of professional contexts.
6. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
7. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
8. Identify and analyze user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing based systems.

PROGRAM SPECIFIC OBJECTIVES (PSO) for M.Sc. Information Technology

Program Specific Objectives (PSOs) are more focused and concrete statements that describe the specific outcomes expected from a Master of Information Technology (M.Sc.) program. These objectives should align with the broader program objectives mentioned earlier.

PSO 1: Analyze and recommend the appropriate IT infrastructure required for the implementation of a project

PSO 2: Design, develop and test software systems for world-wide network of computers to provide solutions to real world problems.

PSO 3: Employ appropriate concepts of problem-solving methods for varied applications

PSO 4: Develop aptitude to meet the challenges and keep themselves abreast of the upcoming trends in the IT industry.

PSO 5: An ability to apply the theoretical concepts and practical knowledge of Information Technology in analysis, design, development and management of information processing system and applications in the interdisciplinary domain

PSO 6: An ability to analyze a problem and identify and define the computing infrastructure and operations requirements appropriate to its solution.

COURSE STRUCTURE AS PER GUIDELINES OF NEP

Illustrative Credit distribution structure for two/ one year M.Sc. Information Technology Programme with Multiple Entry and Exit options for Discipline Specific Course in Information Technology

Semester I							
Course Type	CourseCode	CourseName	Teaching Hrs		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major/Mandatory)-DSC	CIT/MJT/500	Data Structure	2	-	2	-	14
	CIT/MJT/501	Web Technologies	2	-	2	-	
	CIT/MJT/502	Mathematical Foundations	2	-	2	-	
	CIT/MJP/500	Practical based on Data Structure	-	4	-	2	
	CIT/MJP/501	Practical based on Web Technologies	-	4	-	2	
	CIT/MJP/502	Practical baesd on Mathematical Foundations	-	4	-	2	
Skill/Advance Course	CIT/MJP/503-507	Programming-1*	-	4	-	2	2
Discipline Specific Electives - DSE	CIT/DSET/520-523	Elective-1#	2	-	2	-	4
	CIT/DSEP/520-523	Practical baesd on Elective-1#	-	4	-	2	
Research Methodology	CIT/RM/530	Research Methodology	4	-	4	-	4
Total			12	20	12	10	22 Credits

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Semester II

Course Type	CourseCode	CourseName	Teaching Hrs		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major/Mandatory)-DSC	CIT/MJT/550	Advanced Software Engineering	2	-	2	-	14
	CIT/MJT/551	Machine Learning	2	-	2	-	
	CIT/MJT/552	Advanced RDBMS	2	-	2	-	
	CIT/MJP/550	Practical based on Advanced Software Engineering	-	4	-	2	
	CIT/MJP/551	Practical based on Machine Learning	-	4	-	2	
	CIT/MJP/552	Practical based on Advanced RDBMS	-	4	-	2	
Skill/Advance Course	CIT/MJP/553-557	Programming-2*	-	4	-	2	2
Discipline Specific Electives - DSE	CIT/DSET/570-573	Elective-2#	2	-	2	-	4
	CIP/DSEP/570-573	Practical based on Elective-2#	-	4	-	2	
On-Job Training	CIT/OJT/590	On-Job Training / Field Project	-	4	-	4	4
Total			8	28	8	14	22 Credits

24

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Semester III

Course Type	CourseCode	CourseName	Teaching Hrs		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major Mandatory)-DSC	CIT/MJT /600	DataVisualization	2	-	2	-	14
	CIT/MJT /601	Data Mining and Data Warehousing	2	-	2	-	
	CIT/MJT /602	Internet of Things (IoT)	2	-	2	-	
	CIT/MJP /600	Practical based on DataVisualization	-	4	-	2	
	CIT/MJP /601	Practical based on Data Mining and Data Warehousing	-	4	-	2	
	CIT/MJP /602	Practical baesd on Internet of Things (IoT)	-	4	-	2	
Skill/Advance Course	CIT/MJP /603-607	Programming-3*	-	4	-	2	2
Discipline SpecificElectives - DSE	CIT/DSET /620-623	Elective-3#	2	-	2	-	4
	CIT/DSEP /620-623	Practical baesd on Elective-3#	-	4	-	2	
RP	CIT/RP/649	Research Project-I	-	8	-	4	4
Total			8	28	8	14	22 credits

Note: *,#Student is advised to select the any one course from the pool of courses, however horizontal selection of courses tobefollowedat thetimeofselectionofthecourse.

Semester IV

Course Type	CourseCode	CourseName	Teaching Hrs		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major Mandatory)-DSC	CIT/MJT /650	Introduction to Quantum Computing	2	-	2	-	14
	CIT/MJT /651	Artificial Intelligence	2	-	2	-	
	CIT/MJT /652	Big Data Analytics	2	-	2	-	
	CIT/MJP /650	Practical based on Introduction to Quantum Computing	-	4	-	2	
	CIT/MJP /651	Practical based on Artificial Intelligence	-	4	-	2	
	CIT/MJP /652	Practical baesd on Big Data Analytics	-	4	-	2	
Discipline Specific Electives - DSE	CIT/DSET /670 – 673	Elective-4#	2	-	2	-	4
	CIT/DSEP /674- 677	Practical baesd on Elective-3#	-	4	-	2	
RP	CIT/RP/699	Research Project -2	-	12	-	6	6
Total			8	28	8	14	22 credits

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Skill / Advance Course (Programming Elective Group Basket)

Programming Groupm	Programming - 1 CIT/MJP/503-505	Programming-2 CIT/MJP/553-555	Programming -3 CIT/MJP/603-605
Java Group	Core Java	Advance Java	Android
Open Group	Python	Advanced Python	Open Web Programming (PHP)
Web Scripting Group	VB & JavaScript	NodeJS	React

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Discipline Specific Electives - DSE (Elective Group Basket)

Elective Group	Elective 1 CIT/DSET/520-522 CIT/DSEP/520-522	Elective 2 CIT/DSET/570-572 CIT/DSEP/570-572	Elective 3 CIT/DSET/620-622 CIT/DSEP/620-622	Elective 4 CIT/DSET/670-672 CIT/DSEP/670-672
Pattern Analysis & Machine Intelligence	Soft Computing	Fuzzy Systems : Theory, Application & Case Study	Video Processing	Pattern Recognition
Security	Network Security	Cyber Security	Cyber Forensics: Tools, Techniques and Case Studies	Cryptography & Blockchain
Natural Language Processing	Linguistic Fundamentals: Understanding Language Structure and Analysis	Semantics and Pragmatics	Natural Language Processing	AI Chatbot Services and Applications

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Semester I

CourseTitle	Data Structure		
CourseCode(TH)	CIT/MJT/ 500	Course Type	Mandatory
CourseCode(PR)	CIT/MJP/500		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

Any programming language like C, C++, Python etc.

Course Objectives (CO):

- To develop proficiency in the specification, representation, and implementation of DataTypes and Data Structures.
- To be able to carry out the Analysis of various Algorithms for mainly Time and SpaceComplexity.
- To get a good understanding of applications of Data Structures.
- To develop a base for advanced computer science study.

Learning Outcomes :-

- Ability to decide the appropriate data type and data structure for a given problem.
- Ability to select the best algorithm to solve a problem by considering various problemcharacteristics, such as the data size, the type of operations, etc.
- The algorithms as referred above would include various operations on Queues, Stacks, LinkedLists, Trees, Graphs, Sorting, Searching, Hash tables
- Ability to compare algorithms with respect to time and space complexity

Course Outline :-

Unit 1: Linear Data Structures: Arrays, Storage Structure for Arrays, Structures & Arrays of Structures , Stack, Applications of Stacks, Queues, Simulation, Priority Queues, Pointers & Linked Allocation , Linked Linear Lists, Circularly Linked Linear Lists, Doubly Linked Linear Lists, Applications of Linked LinearLists

Unit 2: Nonlinear Data Structures: Trees , Operations on Binary Trees , Storage Representation & Manipulation of Binary Trees, Conversion of General Tree to Binary Trees , Sequential & Other Representation of Trees , Application of Trees - Manipulation of Arithmetic Expression, Graphs and their representation - Matrix Representation of Graphs , Graphic Representation of List Structures , Other Representation of Graphs , Breadth First Search (BFS) , Depth First Search (DFS) , Spanning Trees

Unit 3: Sorting : Introduction , Sorting Techniques -Selection Sort , Bubble Sort , Insertion Sort, Merge Sort , Heap Sort , Quick Sort , Radix Sort

Searching: Sequential Searching, Binary Searching, Search Trees - Binary Search tree, Overview of Balanced trees, Overview of m-ary Trees, Trie Structures, Hash Table Search Methods - Introduction, Hashing Functions, Collision Resolution Techniques

Main Reference Book(s):

1. "An Introduction to Data Structures with Applications", Jean-Paul Tremblay, Paul G.Sorenson, Tata McGraw-Hill, 2nd Edition, (2007)

Suggested Additional Reading:

1. "Data Structures and Algorithm Analysis in C", Mark Allen Weiss, , Pearson Education.
2. "Data Structures: A Pseudo-code Approach with C", Gilberg & Forouzan, , CengageLearning.
3. "Data Structures Via C++: Objects by Evolution", A. Michael Berman, , Oxford Univ.Press (2004)
4. "Fundamentals of Data Structures in C", Horowitz, Sahni, Anderson-Freed, , UniversityPress (2nd edition-2007)
5. "Data Structures Using C & C++", Tenenbaum, PHI.
6. "Data Structures & Algorithms" , A V Aho, J E Hopcroft, J D Ullman, , PearsonEducation (1983).
7. "Sorting & Searching - The Art of Computer Programming" D E Knuth, , Vol. 3, PearsonEducation (1998).
8. "Data structures and algorithms, concepts, Techniques and Applications" ,G. A.V. PAI, ,TMH , 1st Edition (2008)
9. "Algorithm design-foundation, analysis & internet examples", Michel Goodrich, RobertoTamassia, , Wiley
10. "Introduction to Algorithm", Cormen, Leiserson, Rivest, Stein, , PHI (2003), 2nd Edition,
11. "Design and Analysis of Algorithms" Parag Dave & Himanshu Dave, Pearson Education(2008).

CourseTitle	Web Technologies		
CourseCode(TH)	CIT/MJT/ 501	Course Type	Mandatory
CourseCode(PR)	CIT/MJP / 501		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

Student must possess the basics: Markup language HTML,Java Script, and overview ofCSS.

Course Objectives (CO):

- Inthiscourse, students will earn the fundamentals of Web Development and will also get hands-on the recent technologies and tools.
- This course is intended to teach the basics involved in publishing content on the World Wide Web.
- To introduce the fundamentals of Internet, and the principles of web design.
- To construct basic websites using HTML and to build dynamic web pages with validation using Java Scrip, PHP.
- To develop modern interactive web applications using PHP, XML and MySQL

Learning Outcomes (LO):

By the end of the course, students will be able to:

- After studying that subject students would have capability to make own web site and host their own web site on internet. Also students would have enough knowledge about what are the technologies used in internet.
- Students are able to develop a dynamic webpage by the use of java script and XHTML.
- Students will be able to write a well formed / valid XML document.
- Apply the concepts of server side technologies for dynamic web applications
- Implement the web based applications using effective data base access.

CourseOutline:

Unit1: Web Essentials: Clients, Servers, and Communication. The Internet-Basic Internet Protocols World Wide Web-HTTP request message-response message-Web Clients Web Servers-Case Study. Markup Languages: XHTML.An Introduction to HTML History-Versions-Basic XHTML Syntax and Semantics- Some Fundamental HTML Elements-Relative URLs-Lists-tables-Frames-Forms- Creating HTML Documents. Scripting languages: client side and server side.

Unit2: JavaScript and jQuery: Basics of JavaScript and Client-side scripting language, JavaScript syntaxes for variables, functions, branches and repetitions. JavaScript alert, prompt and confirm. Objects in JavaScript, Access/Manipulate web browser elements using DOM Structure, forms and validations, JavaScript events, Basics of jQuery, jQuery syntaxes, jQuery selectors, events, effects, Access/Manipulate web browser elements using jQuery

Unit3: XML: Representing Web Data: XML-Documents and Vocabularies-Versions and Declaration -Namespaces JavaScript and XML Introduction to XML, DTD and Schemas, Well formed, using XML with application.XML, XSL and XSLT. Introduction to XSL, XML transformed simple example, XSL elements, transforming with XSLT. PHP : Starting to script on server side, Arrays, function and forms, Databases : Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, HTTP and Web Server, HTTP Overview, HTTPS/TLS HTTP Prequestmessage- responsemessage- WebClientsWebServers-CaseStudy, Cookies, Server-Side:DynamicContent, Web Content Management Systems (WebCMS).

Reference Books:

1. WebProgramming,buildinginternctapplications,ChrisBates2ndedition,WileyDrcarntech
2. Java Script, D. Flanagan,O'Reilly,SPD.
3. Beginning Web Programming-Jon DuckettW ROX.
4. Web Technologies, UttairK Roy, Oxford UniversityPress

Web/E- References:

1. <https://www.tiitorlalspoint.com/internet technologies/websites development.htm>
2. <https://www.w3sclaoos.coin/html/default.asp>
3. <https://hiedrtartanalytics.nit.edu/>
4. <http://news.nlit.cdli/topic/web-develognJent>

CourseTitle	Mathematical Foundations		
CourseCode(TH)	CIT/MJT/502	CourseType	Mandatory
CourseCode(PR)	CIT/MJP/502		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisite:

Some basic foundations of Matrices and set theory is required to be known to the student before attending this course.

Course Objectives(CO):

- To create the basic foundation of mathematical techniques largely used in Computer Science and Information technology.
- This course covers possibly required mathematics for application development. Study the fundamental concepts of logic, abstract algebra, linear algebra, probability and statistics, graph theory etc.

Learning Outcomes (LO):

By the end the course, students should be able to:

- To relate and solve real life problems using the concepts of sets, relations and functions.
- Will be able to use mathematical foundations in many areas of computer science like algorithms, computer networks, cryptography, etc.

Course Outline:

Unit1: Discrete Mathematics :Sets, Elements of a set , methods of describing a set, types of sets, Operations onsets--union, intersection and difference of sets,Venn diagrams,statement problems,Associative Laws,Distributive laws,DeMorgans laws, duality, partitioning of a set. Relation -Basic definition of relation and types of relations, graphs of relations, properties of relations, recurrence relations, Matrix representation of a relation, Eigen values and their Eigen vectors, concepts of logic.

Unit2: Descriptive and Inferential Statistics, Measures of Central Tendency, Mean, Median, Mode, Other Averages, Measures of Dispersion, Range, Mean Deviation, Standard Deviation, Measures of Skewness, Kurtosis, Measures of Relationship, Covariance, Sampling and Statistical Inference, Parameter and Statistic, Sampling and Non-sampling Errors, Sampling Distribution, Sampling Distribution of Mean, Sampling Distribution of Proportion, Student's t- Distribution , Statistical Inference, Point Estimation, Interval Estimation, Sample Size and its Determination ,Tests of Significance , Analysis of Variance.

Unit 3:Linear Regression Analysis, Dependent and Independent Variables, Simple Linear Regression Model, Least Squares Estimation, Multiple Linear Regression Model, Least Squares Estimation, Combinatorics: Basic of Counting, Permutations, Permutations with Repetitions. Introduction to graph theory.

Recommended Reference Book

1. Kenneth H. Rosen (Indian Adaptation By Kamala Krithivasan), Discrete Mathematics And Its Applications With Combinatorics And Graph Theory, seventh edition, McGraw Hill Education. (Unit 1)
2. C. L. Liu, D. P. Mohapatra, "Elements of Discrete Mathematics" 3rd edition, McGraw Hill, 2008. (Unit 1)
3. Kothari C. R. & Garg Gaurav, (2019), Research Methodology Methods & Techniques (fourth Edition), New Age International Publishers, New Delhi. (Unit 2-5)
4. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rd Edition, Tata McGraw Hill.
5. Kolman and Busby—Discrete Mathematical Structures for Computer Sciences Pm.

Course Title	Research Methodology (Information Technology)		
Course Code	CIT/RM/530	Course Type	Mandatory
Credits	4 Credits	Contact Hours (TH)	4 Hrs/ Week

Prerequisites:

- Basic understanding of computer science concepts
- Familiarity with programming languages (e.g., Python, R, or MATLAB)
- Knowledge of introductory statistics

Course Objectives (CO):

- To introduce students to the fundamental principles of research methodology in computer science.
- To provide students with a comprehensive understanding of computational and statistical methods used in computer science research.
- To equip students with the skills to collect, pre-process, analyse, and interpret data for research purposes.
- To enable students to apply appropriate statistical techniques for hypothesis testing and inference.
- To familiarize students with machine learning algorithms and their applications in computer science research.
- To develop students' ability to design and analyse experiments in the context of computer science research.
- To enhance students' critical thinking and problem-solving skills through case studies and practical assignments.

Learning Outcomes (LO):

By the end of the course, students will be able to:

- Formulate research questions and select appropriate research methodologies.
- Collect, pre-process, and analyse data using computational and statistical techniques.
- Apply hypothesis testing and statistical inference to draw meaningful conclusions from data.
- Build and evaluate regression and machine learning models for predictive analysis.
- Design and analyse controlled experiments to investigate research hypotheses.
- Demonstrate an understanding of real-world applications of computational and statistical methods in computer science research.
- Critically evaluate research studies and identify strengths and limitations in their methodology.

Course Outline:

Unit 1: Introduction to Research Methodology- Overview of research methodology in computer science, Research design and problem formulation, Literature review and identifying research gaps, Ethical considerations in research

Unit 2: Data Collection and Pre-processing - Data collection techniques: surveys, interviews, observations, etc. Data pre-processing and cleaning, Handling missing data and outliers, Exploratory data analysis, Case study: Visualizing and summarizing real-world datasets

Unit 3: Statistical Analysis - Descriptive statistics: measures of central tendency, dispersion, etc., Hypothesis testing and statistical significance, Parametric and non-parametric tests, Analysis of variance (ANOVA) and regression analysis, Case study: Applying statistical inference techniques to analyze experimental data

Unit 4: Computational Analysis - Introduction to machine learning algorithms, Supervised and unsupervised learning techniques, feature selection and dimensionality reduction, Evaluation metrics for Computational Methods and Techniques, Case study: Building a computational model for classification or clustering

Unit 5: Presenting Research Findings-Effective data visualization techniques, Scientific writing and report preparation, Presenting research findings in conferences and journals, Peer review process and publication ethics.

Assessment Methods:

- Assignments and quiz to assess understanding of concepts and techniques
- Course project report and presentation evaluation
- Participation in class discussions and group activities

Recommended Reference Books:

1. "Research Methodology: A Step-by-Step Guide for Beginners" by Ranjit Kumar
2. "Designing and Conducting Mixed Methods Research" by John W. Creswell and Vicki L. Plano Clark "Statistical Methods for Computer Science" by Walter D. Wallis
3. "Pattern Recognition and Machine Learning" by Christopher M. Bishop "Data Science for Business" by Foster Provost and Tom Fawcett
4. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
5. "Visualization Analysis and Design" by Tamara Munzner

Semester-II

CourseTitle	Advanced Software Engineering		
CourseCode(TH)	CIT/MJT/ 550	Course Type	Mandatory
CourseCode(PR)	CIT/MJP/550		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites: The student must aware of software development paradigms.

Course Objectives (CO):

- To understand the concept of Software engineering an Phases.
- To gain knowledge of the Software requirement Analysis and Design concepts.
- To understand software testing approaches for various Application.
- Gain insights into emerging trends in software engineering and their practical implications.
- Develop critical thinking and problem-solving skills related to complex software development challenges.

Learning Outcomes (LO):

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

- Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development.
- Deliver quality software products by possessing the leadership skills as an individual or contributing to the team development and demonstrating effective and modern working strategies by applying both communication and negotiation management skill.
- Plan and deliver an effective software engineering process, based on knowledge of widely used development lifecycle models.
- Formulate a testing strategy for a software system, employing techniques such as unit testing, test driven development and functional testing.

Course Outline:

Unit 1: Software Engineering, Software characteristics, Applications of software, Software Development activities. Software Lifecycle models - Classical waterfall Iterative waterfall, V-Model, Incremental Model , RAD Model, Spiral model, Prototype Model. Software Requirements and Analysis: System Engineering, Product Engineering: Characteristics of a Good SRS, Requirement Analysis, Principal, Software prototyping, Specification and its review. Analysis modeling: data modeling, mechanics for structured analysis, system analysis tools and techniques, Data Flow Diagram, ER- Diagrams. Data Dictionary.

Unit 2: The Design Process, Design Concepts, Design Model. Architectural Design: Software Architecture, Architectural Genres, Architectural Styles Architectural Design, Assessing Alternative Architectural Designs. Component Level Design: Designing Class-Based Components, Conducting Component-Level Design, Cohesion and Coupling. User Interface Design: The Golden Rules, Interface Analysis and Design, Interface Analysis Interface Design Steps. Strategic approach to software testing -Introduction to Testing ,Verification and Validation, A software testing strategy ,Criteria for completion of testing, Unit testing – Black box testing– White box testing – Integration, Validation testing System testing– Regression testing, Debugging process and approaches.

Unit 3: Web Engineering: Engineering Layers, Engineering Process, Formulating web based systems, Planning, Team, Project Management, Metrics for Web Engineering and WebApps, Analysis model for WebApps, Content Model, Interaction Model, Functional model, Configuration model, Navigation analysis, WebApp Design and Testing. Cleanroom software engineering- Clean Room approach, functional specification, Cleanroom design, Cleanroom testing Component based Development: The CBSE Process, Domain engineering, Component based development, Classifying and Retrieving Components, Economics of CBSE. Agile Development-Agile practices, extreme programming, planning, testing, refactoring, Agile design basics. Software process models and metrics for evolving technologies.

Recommended Reference Book

- Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India.
- Pankaj Jalote, An integrated approach to Software Engineering, Springer/Narosa.
- Roger S. Pressman, Software Engineering – A Practitioners Approach, 4th/7th Edition, McGraw Hill, International Education.
- Ian Sommerville, Software Engineering, Addison-Wesley.
- Heinemann, G.T., and Councill, W.T., "Component-Based Software Engineering: Putting the Pieces Together", Pearson Higher Education/Addison Wesley
- Pressman, R. S. and Lowe, D., "Web Engineering: A Practitioner's Approach", Special Indian Edition, Tata McGraw-Hill.

Barrios Bier, Software Testing Techniques, 2nd Edition, Van N Ostrand Reinhold.

CourseTitle	Machine Learning		
CourseCode(TH)	CIT/MJT/ 551	Course Type	Mandatory
CourseCode(PR)	CIT/MJP/ 551		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of probability and statistics
- Familiarity with programming (Python preferred)
- Understanding of linear algebra concepts
- Some exposure to calculus

Course Objectives (CO):

- To provide students with a comprehensive understanding of statistical machine learning techniques and algorithms.
- To equip students with the knowledge and skills necessary to apply statistical machine learning to real-world problems.
- To develop critical thinking and analytical skills for model evaluation, selection, and optimization.
- To introduce students to popular machine learning frameworks and libraries for implementation.

Learning Outcomes (LO):

By the end of this course, students should be able to:

- Understand the fundamental concepts and principles of statistical machine learning.
- Apply supervised and unsupervised learning algorithms to analyze and interpret data.
- Evaluate and compare the performance of different machine learning models.
- Implement machine learning algorithms using programming languages and framework.
- Apply statistical machine learning techniques to solve real-world problems in various domains.
- Demonstrate the ability to interpret and present the results of machine learning models effectively.

Course Outline:

Unit 1: Introduction to Statistical Machine Learning - Overview of statistical machine learning, Supervised and unsupervised learning, Model evaluation and selection. Regression Analysis - Linear regression, Polynomial regression, Regularization techniques (e.g., Ridge, Lasso), Case study: Predicting housing prices

Unit 2: Classification Methods - Logistic regression, Naive Bayes classification, Decision trees and ensemble methods (e.g., Random Forest, Gradient Boosting), Case study: Spam email classification.

Unsupervised Learning- Clustering algorithms (e.g., K-means, Hierarchical clustering), Dimensionality reduction techniques (e.g., PCA, t-SNE) Case study: Customer segmentation

Unit 3: Deep Learning Fundamentals - Neural networks and deep learning architectures, Convolutional Neural Networks (CNNs) for image analysis, Recurrent Neural Networks (RNNs) for sequential data, Case study: Image classification using CNNs.

Case studies

1. Sentiment Analysis of Social Media Data: Analyzing Twitter data to classify sentiment (positive, negative, neutral) using various machine learning models.
2. Image Classification with Convolutional Neural Networks: Building a model to classify images from the CIFAR-10 dataset using CNNs.
3. Movie Recommender System: Developing a recommendation engine using collaborative filtering techniques to suggest personalized movie recommendations.
4. Credit Card Fraud Detection: Building a fraud detection model using supervised learning algorithms to identify fraudulent transactions.
5. Time Series Forecasting: Predicting stock prices or weather patterns using time series forecasting techniques such as ARIMA or LSTM models.

Recommended Reference Books:

1. "Pattern Recognition and Machine Learning" by Christopher Bishop
2. "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville,

Course Title	Advanced RDBMS		
Course Code (TH)	CIT/MJT / 552	Course Type	Mandatory
Course Code (PR)	CIT/MJP / 552		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites :

- 1) The student should have basic knowledge of database concepts.
- 2) The student should have basic knowledge of SQL.

Course Objectives (CO):

- Understand the concepts of Transactions, concurrency control, recovery, security and integrity
- Understand various database system architectures
- Learn to work with PL/SQL
- Understand how to handle errors and use triggers

Learning Outcomes (LO):

- Use the concepts of Transactions, concurrency control, recovery, Security and integrity
- Elaborate on various database system architectures
- Work with PL/SQL
- Able to handle errors and use triggers

CourseOutline:

Unit 1: Transactions: ACID properties, states of transaction, Concurrent executions, concepts of serializability and recoverability. Concurrency control: Overview of Locking, modes of locking: shared & exclusive, 2 phase locking protocol, time stamping, Timestamp-ordering protocol, validation- based protocol, multi-versioning, Deadlock Handling.

Recovery: transaction failure classification, stable storage implementation, log- based recovery, Shadow paging, recovery with concurrent transactions, checkpoints & rollback. Security & Integrity: security measures for protection of data at various levels, authorization, views, granting of privileges, security specifications in SQL, encryption.

Unit 2: Introduction to PLSQL, PL/SQL syntax, block structure – declarative part, executable part, exception handling part, variable declaration using %type, % rowtype, if statements, looping structures, cursors & its types, cursor attributes, nesting of cursors, parameterized cursors, error handling in SQL. Locks, implicit locking, levels of locks, explicit locking, select for update statement, using lock table statement.

Unit 3: Error handling: user named exception handlers for i/o validation and business rule validation. Stored Procedures and Functions: creating a stored procedure or function, syntax for declaration, execution and exception handling parts, advantages of using procedure or function. Deleting a procedure or function. Database Triggers: Introduction, use, database triggers v/s procedures, database triggers v/s declarative integrity constraints, how to apply triggers. Types of triggers, Creating a trigger, deleting trigger. User defined error messages.

RecommendedReferenceBooks:

1. "Database System Concepts", Abraham Silberschatz, Henry Korth, S. Sudarshan, McGraw Hill
2. "Database Management System", Rajesh Narang, PHI
3. "An introduction to database system ", C. J. Date
4. "An Introduction to Database System", Bipin C. Desai
5. "Database management system", Ramakrishnan Gehrke.

Semester III

CourseTitle	Data Visualization		
CourseCode(TH)	CIT/MJT / 600	Course Type	Mandatory
CourseCode(PR)	CIT/MJT /600		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

- Basic understanding of statistics and data analysis concepts
- Familiarity with spreadsheet software (e.g., Microsoft Excel, Google Sheets)
- Basic programming knowledge (preferred but not mandatory)

Course Objectives (CO):

- Understand the fundamental principles and techniques of data visualization.
- Gain proficiency in selecting appropriate visualization techniques for different types of data.
- Develop skills in creating visually appealing and effective data visualizations.
- Learn to design interactive and engaging visualizations to facilitate data exploration.
- Apply data visualization concepts to real-world case studies and datasets.
- Enhance story telling abilities by incorporating data visualization techniques.
- Cultivate ethical awareness and responsibility in data visualization practices.

Learning Outcomes (LO):

By the end of the course, students should be able to:

- Demonstrate a comprehensive understanding of data visualization principles and techniques.
- Evaluate and select appropriate visualization techniques based on data types and objectives.
- Design and create visually compelling and informative data visualizations.
- Develop interactive visualizations to facilitate data exploration and user engagement.
- Apply data visualization skills to analyze and present complex data sets effectively.
- Incorporate storytelling elements into data visualizations to convey insights clearly.
- Recognize and address ethical considerations in data visualization practices.

Course Outline:

Unit 1: Introduction to Data Visualization - Definition and importance of data visualization, Types of data visualization techniques, Overview of data visualization tools and software. Principles of Effective Data Visualization - Gestalt principles and visual perception, Color theory and best practices, Layout and composition guidelines, Data storytelling and narrative techniques

Unit 2: Data Types and Visualization Technique - Visualizing numerical data: line charts, bar charts, scatter plots, etc, visualizing categorical data: pie charts, stacked bar charts, treemaps, etc, Visualizing temporal data: time series plots, heatmaps, calendars, etc, Visualizing spatial data: maps, choropleth maps, cartograms, etc. Interactive Data Visualization - Introduction to interactive visualization tools, Adding interactivity using tooltips, filters, and selection, Creating interactive dashboards and exploratory visualizations, designing for user engagement and ease of use.

Unit 3: Advanced Data Visualization Techniques - Network visualization and graph-based data, Hierarchical and tree-based visualizations, 3D and multidimensional visualizations, Geospatial and geo-visualization techniques. Data Visualization and Storytelling - Communicating insights through effective data visualization, designing visual narratives and story arcs, integrating text, annotations, and interactivity into visualizations, Presenting data visualizations to diverse audiences. Data Visualization Ethics and Best Practices - Ethical considerations in data visualization, Data accuracy, integrity, and representation, Avoiding bias and misleading visualizations, Responsible data storytelling and interpretation.

Case Studies:

1. Visualizing Global Health Data: Analyzing and presenting health indicators across countries to identify patterns and disparities.
2. Interactive Sales Dashboard: Designing an interactive dashboard to explore sales data and identify trends, regions, and product performance.
3. Network Analysis: Visualizing social networks or organizational structures to reveal connections and influence patterns.
4. Geospatial Data Visualization: Mapping and analyzing geographic data, such as population density, distribution of resources, or climate patterns.
5. Time Series Visualization: Analyzing temporal data, such as stock prices or weather patterns, to identify trends and make predictions.

Recommended Reference Book

"Data Visualization: A Practical Guide" by Andy Kirk, SAGE Publications Ltd. 2021

CourseTitle	Data Mining and Data Warehousing		
CourseCode(TH)	CIT/MJT /601	Course Type	Mandatory
CourseCode(PR)	CIT/MJP /601		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of programming concepts
- Familiarity with Python programming language(recommended)
- Understanding of fundamental statistical concepts
- Basic knowledge of data bases and SQL

Course Objectives (CO):

- Understand the concepts and principles of data mining and warehousing
- Gain proficiency in using Python for data mining and warehousing tasks
- Learn various data pre-processing techniques for preparing data for analysis
- Explore different data mining algorithms and their applications
- Understand the architecture and design principles of data warehousing
- Apply data visualization techniques to communicate insights effectively
- Analyse real-world case studies to gain practical experience in data mining and warehousing

Learning Outcomes (LO):

By the end of the course, students will be able to:

- Understand the role and significance of data mining and warehousing in decision-making processes.
- Apply Python programming language and relevant libraries for data mining and warehousing tasks.
- Pre-process and clean data effectively, including handling missing values, outliers, and data transformation.
- Apply various data mining techniques such as association rule mining, clustering, classification, regression, and text mining.
- Design and implement a data warehouse, including ETL processes and dimensional modeling.
- Utilize OLAP tools for data analysis and reporting.
- Create compelling data visualizations and interactive dashboards.
- Analyse real-world case studies to solve business problems using data mining and warehousing techniques.

Course Outline:

Unit 1: Introduction to Data Mining - Overview of data mining and its applications, Data pre-processing: cleaning, integration, transformation, and reduction, Exploratory data analysis and visualization techniques, Introduction to Python libraries for data mining (e.g., Pandas, NumPy, Matplotlib). **Association Rule Mining** - Apriori algorithm and association rule generation, Evaluation metrics for association rules, Practical applications and case studies of association rule mining

Unit 2: Clustering - Introduction to clustering algorithms (e.g., k-means, hierarchical clustering), Evaluation metrics for clustering, Practical applications and case studies of clustering. **Classification** - Introduction to classification algorithms (e.g., decision trees, Naïve Bayes, logistic regression), Evaluation metrics for classification, Practical applications and case studies of classification,

Unit 3: Introduction to Data Warehousing - Overview of data warehousing and its components, Data modelling: star schema, snowflake schema, ETL processes: extraction, transformation, and loading of data, Practical applications and case studies of data warehousing. **OLAP and Data Cube** - Introduction to OLAP and its benefits, Data cube representation and operations, OLAP tools and techniques, Practical applications and case studies of OLAP. **Case Studies** – a) Retail Market Basket Analysis:

Analysing customer purchase patterns to identify associations between products for targeted marketing b) Customer Segmentation: Clustering customers based on demo graphic and behavioural data to tailor marketing campaigns c) Loan Default Prediction: Using classification techniques to predict the likelihood of loan default based on historical data.

Reference Books:

"Data Mining: Concepts and Techniques "by Jiawei Han, Micheline Kamber, and JianPei."Python for Data Analysis "by Wes McKinney.

"Data Warehousing Fundamentals"by Paulraj Ponniah.

"The Data Warehouse Tool kit: The Definitive Guide to Dimensional Modeling"by Ralph Kimball and Margy Ross.

Course Title	Internet of Things (IoT)		
Course Code (TH)	CIT/MJT/ 602	Course Type	Mandatory
Course Code (PR)	CIT/MJP /602		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of computer networks and protocols
- Familiarity with programming concepts (preferably in a language like Python or Java)
- Understanding of data management and analytics principles

Course Objectives (CO):

- To provide an understanding of the fundamental concepts and components of the Internet of Things (IoT)
- To explore various technologies and protocols used in IoT networks and devices
- To introduce methods for acquiring, processing, and analyzing IoT data
- To examine real-world IoT applications and case studies in different domains
- To address the security and privacy challenges associated with IoT deployments
- To enable students to design and develop IoT solutions using appropriate platforms and frameworks

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Define and explain the key concepts, components, and architecture of the Internet of Things
- Identify and evaluate different IoT devices, sensors, and communication technologies
- Design and implement IoT data acquisition and processing systems
- Analyze and interpret IoT data for decision-making purposes
- Develop IoT applications using relevant platforms and frameworks
- Assess security risks and implement appropriate measures for securing IoT deployments
- Analyze and discuss real-world case studies of successful IoT implementations in various domains

Course Outline:

Unit 1: Introduction to IoT - Definition, history, and evolution of IoT, Key components of an IoT system, IoT ecosystem and stakeholders. IoT Architecture and Protocols-IoT network architectures (centralized, decentralized, hybrid), Communication protocols (MQTT, CoAP, HTTP, etc.), Sensor networks and data aggregation. IoT Connectivity Technologies - Wireless communication (Wi-Fi, Bluetooth, Zigbee, LoRa, etc.), Cellular technologies (2G, 3G, 4G, 5G), Edge computing and for computing

Unit2:IoT Data Management and Analytics-Datacollection, storage, and processing in IoT, Big Data analytics and machine learning for IoT, Data security and privacy considerations. IoT Applications in Smart Home and Cities-Smart home automation systems, Intelligent transportation systems, Energy management and environmental monitoring. IoT in Healthcare and Wearable Devices-Remote patient monitoring, Smart health care systems, Wearable technology and healthcare applications

Unit 3: Industrial IoT (IIoT) and Smart Manufacturing - Industrial automation and control system, Predictivemaintenance and asset tracking, Supply chain management and logistics, Future Trends and Challenges in IoT –Emerging trends in IoT(AIintegration, edge intelligence,etc), Ethical considerations and societal impact of IoT,Open research problems and future directions. CaseStudies a) Smart agriculture and precision farming b) Smart retail and inventory management, c)Smart buildings and infrastructure, d) IoT- enabled environmental monitoring

Recommended Reference Books:

1. "Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz
2. "Internet of Things: Principles and Paradigms" by Rajkumar Buyya, Amir Vahid Dastjerdi
3. "Designing Connected Products: UX for the Consumer Internet of Things" by Claire Rowland, Elizabeth Goodman, Martin Charlier, AnnLight
4. "Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security" byPerry Lea
5. "The Fourth Industrial Revolution"by Klaus Schwab

Semester IV

CourseTitle	Introduction to Quantum Computing		
CourseCode(TH)	CIT/MJT/ 650	Course Type	Mandatory
CourseCode(PR)	CIT/MJP/ 650		
Credits	4Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of linear algebra
- Familiarity with probability theory
- Understanding of classical computing concepts

Course Objectives (CO):

- Introduce the principles of quantum mechanics and the irrelevance to computing.
- Develop an understanding of quantum gates, circuits, and algorithms.
- Explore the fundamentals of quantum error correction and fault tolerant computation.
- Examine applications of quantum computing in areas such as simulation, optimization, and cryptography.

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Understand the foundational principles of quantum mechanics and their application to computing.
- Design and analyze basic quantum circuits and algorithms.
- Evaluate the potential advantages and limitations of quantum computing in various domains.

Course Outline:

Unit1: Introduction to Quantum Mechanics-Historical overview of quantum mechanics, Principles of superposition and entanglement, Quantum states and qubits, Measurement and quantum interference, Quantum Gates and Circuits - Single-qubit gates (Pauli gates, Hadamard gate, phase gates), Multiple-qubit gates (CNOT, SWAP, Toffoli), Quantum circuits and circuit simplification, Universal gates and quantum computing models

Unit 2: Quantum Algorithms - Quantum parallelism and the Deutsch-Jozsa algorithm, Grover's search algorithm, Shor's factorization algorithm, Quantum simulation and the Quantum Phase Estimation algorithm, Quantum Error

Correction and Noise-Sources of errors in quantum computing, Quantum error correction codes, Quantum noise and decoherence, Error mitigation techniques

Unit 3: Quantum Hardware and Technologies - Different platforms for quantum computing (superconducting qubits, trapped ions, topological qubits), Quantum gates and operations in hardware, Quantum processor architectures and scalability challenges, Quantum software development frameworks and tools. Case Studies in Quantum Computing a) Quantum machine learning and optimization, b) Quantum cryptography and secure communication c) Quantum chemistry simulations, d) Quantum finance and portfolio optimization

Recommended Reference Books:

"Quantum Computing for Computer Scientists" by Noson S. Yanofsky and Mirco A. Mannucci

"Quantum Computing: A Gentle Introduction" by Eleanor Rieffel and Wolfgang Polak "Quantum Computation and Quantum Information" by Michael A. Nielsen and Isaac L. Chuang

"Quantum Computing: From Linear Algebra to Physical Realizations" by Mikio Nakahara and Tetsuo Ohmi "
Quantum Computing: An Applied Approach" by Jack D. Hidary

Course Title	Artificial Intelligence		
Course Code (TH)	CIT/MJT / 651	Course Type	Mandatory
Course Code (PR)	CIT/MJP / 651		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic programming knowledge (Python preferred)
- Understanding of linear algebra and calculus
- Familiarity with statistics and probability theory

Course Objectives (CO):

- Gain a practical understanding of the core concepts and technique in Artificial Intelligence.
- Develop the skills to implement and train machine learning models for various tasks.
- Explore advanced topics in deep learning, natural language processing, computer vision, and reinforcement learning.
- Apply AI techniques to real-world problems through case studies.
- Understand the ethical considerations and challenges associated with AI implementation.

Learning Outcomes (LO):

By the end of the course, students will be able to:

- Implement and evaluate machine learning models using popular libraries and frameworks.
- Build and train neural networks for various AI tasks.
- Apply NLP technique to analyse and process textual data.
- Utilize computer vision algorithms for image analysis and object recognition.
- Understand the principles of reinforcement learning and develop RL-based agents.
- Analyse and interpret AI case studies in different domains.
- Demonstrate an awareness of ethical considerations in AI and the impact of AI on society.

Course Outline:

Unit1: Introduction to Artificial Intelligence- Definition and history of AI, Types of AI systems , AI applications and impact on society. Revisiting Machine Learning Fundamentals - Supervised, unsupervised, and reinforcement learning, feature engineering and data preprocessing, Evaluation metrics

Unit 2: Neural Networks and Deep Learning - Introduction to neural networks, Convolution Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep learning frameworks (e.g., TensorFlow, PyTorch), Reinforcement Learning- Markov Decision Processes (MDPs), Q- learning and policy gradients, Deep reinforcement learning.

Unit 3: Future Trends and Career Opportunities in AI - Emerging AI technologies (e.g., Generative AI, Quantum AI) , AI Case studies – a) Natural Language Processing (NLP) -Sentiment analysis and language generation, b) Computer Vision and Image Processing- Autonomous vehicles, c) Recommender Systems- Personalized recommendation systems, d) AI in Healthcare - AI-assisted diagnosis, e) AI in Manufacturing and Logistics – AI in manufacturing processes, f) AI in Education- Personalized education with AI, g) AI in Governance

#use of additional learning resource is highly appreciated.

Recommended Reference Books:

1. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
2. "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper "Computer Vision: Algorithms and Applications" by Richard Szeliski
3. "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto
4. "The Hundred-Page Machine Learning Book" by Andriy Burkov

Course Title	Big Data Analytics		
Course Code (TH)	CIT/MJT / 652	Course Type	Mandatory
Course Code (PR)	CIT/MJP / 652		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of statistics and probability
- Familiarity with programming concepts (preferably Python or R)
- Understanding of database concepts and SQL
- Some exposure to data analysis or data-driven decision-making

Course Objectives (CO):

- To provide students with a comprehensive understanding of data analytics concepts, techniques, and tools.
- To develop practical skills for data collection, pre-processing, and exploratory analysis.
- To introduce students to various data mining techniques and their applications in solving real-world problems.
- To equip students with the knowledge and skills required for predictive modelling and machine learning in data analytics.
- To explore big data analytics concepts and scalable data processing frameworks.

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Understand the fundamental concepts and principles of data analytics.
- Collect, preprocess, and analyze data using various techniques and tools.
- Apply exploratory data analysis techniques to gain insights and visualize data effectively.
- Utilize different data mining techniques for pattern discovery, clustering, and classification.
- Understand the challenges and opportunities associated with big data analytics.

Course Outline

Unit 1: Introduction to Data Analytics - Understanding the importance of data analytics, Overview of data analytics lifecycle, Introduction to data analysis tools and technologies. Data Preparation and Cleaning - Data collection and pre-processing techniques, Handling missing data and outliers, Data transformation and feature engineering. Exploratory Data Analysis (EDA) - Descriptive statistics and data visualization, Univariate and multivariate analysis techniques, Exploring relationships and patterns in data.

Unit 2: Statistical Analysis - Hypothesis testing and confidence intervals, Parametric and non-parametric tests, Regression analysis and correlation. Predictive Analytics - Introduction to predictive modelling, Regression and classification algorithms, Model evaluation and selection. Time Series Analysis - Understanding time series data, Trend analysis and seasonality, Forecasting techniques. Clustering and Dimensionality Reduction - Unsupervised learning algorithms, K-means clustering and hierarchical clustering, Principal Component Analysis (PCA)

Unit 3: Case Studies a) Text Mining and Sentiment Analysis - Basics of text mining, Text pre-processing and feature extraction, Sentiment analysis techniques, b) Network Analysis - Introduction to network theory, Social network analysis, Network visualization and metrics, c) Analysing real-world datasets and scenarios (data can be downloaded from <https://www.kaggle.com/datasets>), d) Applying data analytics techniques to solve problems on the Kaggle Dataset (<https://www.kaggle.com/datasets>), e) Discussion and interpretation of case study results

Reference Book

1. "Data Analytics for Beginners" by John Smith
2. "Data Science for Business" by Foster Provost and Tom Fawcett "Python for Data Analysis" by Wes McKinney
3. "Statistics for Data Science" by James Miller
4. "Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die" by Eric Siegel "Forecasting: Principles and Practice" by Rob J. Hyndman and George Athanasopoulos "Pattern Recognition and Machine Learning" by Christopher M. Bishop

Course Contents for Skill / Advance Course (Programming Elective Group Basket)

Programming Group	Programming-1 CIT/MJP/503-505	Programming-2 CIT/MJP /553-555	Programming-3 CIT/MJP /603-605
Java Group	Core Java	Advance Java	Android
Open Group	Python	Advanced Python	OpenWeb Programming (PHP)
WebScripting Group	VB & Java Script	Node JS	React

Note: *Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Programming Group: Java Group

Course Title	Core Java		
Course Code (PR)	CIT/MJP /503	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisite:

No prior programming experience is required for this course. However, a basic understanding of computer systems and general familiarity with using computers is recommended.

Course Objectives (CO):

- Understand the fundamental concepts and principles of the Java programming language.
- Gain proficiency in writing, compiling, and executing Java programs.
- Develop object-oriented programming skills and design reusable code.

Learning Outcome (LO):

By the end of this course, students will be able to:

- Write Java programs to solve simple to complex programming problems.
- Understand and apply object-oriented programming concepts effectively.
- Utilize the Java Collections Framework to manage and manipulate data efficiently.

Course Outline:

Unit 1: Introduction to Java - History and features of Java, Java development environment setup, Basic syntax, data types, and variables, Operators and expressions, Control flow statements (if-else, loops), Object-Oriented Programming (OOP) Basics - Classes and objects, Encapsulation, Inheritance, and Polymorphism, Method overloading and overriding, Constructors and static members, Access modifiers

Unit 2: Advanced OOP Concepts - Abstract classes and interfaces, Packages and access control, Exception handling, Enumerations, Inner classes, Java Collections Framework - Introduction to collections, ArrayList, LinkedList, and Vector, HashSet, LinkedHashSet, and TreeSet, HashMap, LinkedHashMap, and TreeMap, Iterators and foreach loop, File Handling and I/O - File class and file operations, Character streams and byte streams, Reading and writing files, Serialization and deserialization, Working with directories, Multithreading - Understanding threads, Creating and running threads, Synchronization and thread safety, Thread communication and coordination, Threadpools and executors

Unit 3: GUI Programming with Applet and Swing - Life Cycle of Applets & Swings components, Creating UI components - Applet & Swings (buttons, labels, text fields, etc.), Event-driven programming, Layout managers, Creating Dialogs and message boxes, Swing Layouts. Revisiting Databases and SQL - Understanding databases and database management systems, Introduction to SQL (Structured Query Language), Performing basic database operations using SQL. Database Integration with Java (JDBC) - Overview of JDBC (Java Database Connectivity), Connecting to databases using JDBC, Executing SQL queries and retrieving results in Java, Advanced JDBC Concepts - Prepared statements and parameterized queries, Batch processing and transaction management, Handling resultsets and metadata

Case Studies and Project Work: Case studies covering real-world scenarios, implementing case study solutions using Core Java. Following Case studies are required to be addressed by the students during their laboratory work.

CaseStudy a)Building a Contact Management System-Analyzing requirements, designing the database schema, Implementing the system using Core Java, databases, and Swing, b) Creating an Inventory Management Application-Understanding the business case, Designing the database structure Developing the application with Core Java, databases, and Swing

Capstone Project: Final Capstone project work incorporating multiple concepts learned by the student during the course work with Code optimization and best practices.

Recommended Reference Books:

1. "Head First Java "by Kathy Sierra and Bert Bates
2. "Core Java, Volume I--Fundamentals"by Cay S. Horstmann
3. "Effective Java"by Joshua Bloch
4. "Java: The Complete Reference"by Herbert Schildt
5. "Thinking in Java"by Bruce Eckel

Course Title	Advance Java		
Course Code (PR)	CIT/MJP /553	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-I Elective Basket of Programming Group at Semester 1		

Prerequisite:

- Solid understanding of core Java programming concepts
- Familiarity with web development basics (HTML, CSS, and JavaScript)
- Knowledge of relational databases and SQL

Course Objectives (CO):

- Develop a deep understanding of advanced Java web development concepts
- Gain proficiency in using Servlets, Struts, and Hibernate frameworks
- Learn to design and build scalable and maintainable web applications
- Acquire hands-on experience in implementing MVC architecture
- Develop skills in database integration and ORM techniques

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Design and develop advanced Java – based web applications
- Effectively use Servlets, Struts, and Hibernate frameworks in web development projects
- Apply industry – standard design patterns and best practices
- Implement secure authentication and session management
- Create efficient database-driven web applications
- Demonstrate proficiency in integrating and utilizing multiple frameworks

Course Outline:

Unit 1: Introduction to Advanced Java Web Development - Overview of Java Servlets, Struts, and Hibernate, Comparison of different web frameworks, Understanding the MVC (Model-View-Controller) architecture. Servlets: Building Dynamic Web Applications - Servlet life cycle and request handling, Handling form data and request parameters, Session management and authentication, Servlet filters and listeners, Error handling and exception management

Unit 2: Struts Framework: Structured Web Application Development - Introduction to Struts framework and its components, Configuring Struts and defining action mappings, working with forms and validation, managing database operations using Struts, Implementing security and authentication in Struts. **Hibernate: Object-Relational Mapping (ORM)** - Introduction to Hibernate and ORM concepts, Configuring Hibernate with different database systems, Mapping Java objects to database tables, Performing CRUD operations using Hibernate, Querying data using Hibernate Query Language (HQL)

Unit 3: Integration of Servlets, Struts, and Hibernate - Leveraging the power of Servlets, Struts, and Hibernate together, building a complete end-to-end web application, Implementing layered architecture and separation of concerns, Handling transactions and database operations. Case Studies a) Building an e-commerce platform using Servlets, Struts, and Hibernate, b) Develop in a social networking application with advanced Java web technologies, c) Creating a banking system with secure authentication and transaction handling, d) Implementing a content management system (CMS) using Java web frameworks

Recommended Reference Books:

1. "Head First Servlets and JSP" by Bryan Basham, Kathy Sierra, and Bert Bates
2. "Struts2 in Action" by Don Brown, Chad Davis, Scott Stanlick, and Ted Husted
3. "Hibernate in Action" by Christian Bauer and Gavin King
4. "Pro Spring MVC: With Web Flow" by Marten Deinum, Koen Serneels, and Colin Yates
5. "Java Persistence with Hibernate" by Christian Bauer and Gavin King

Course Title	Android		
Course Code (PR)	CIT/MJP /603	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-I Elective Basket of Programming Group at Semester 1		

Prerequisite:

- Basic knowledge of Java programming language
- Familiarity with object-oriented programming concepts

Course Objectives (CO):

- Understand the fundamental concepts and architecture of the Android platform.
- Gain proficiency in developing Android applications using Java.
- Acquire knowledge of various Android components and their functionalities.
- Learn to design user-friendly and visually appealing Android interfaces.
- Develop skills in handling data storage, network communication, and multimedia integration in Android applications.
- Apply best practices and coding standards for developing high-quality Android apps.
- Gain hands-on experience by working on real-world case studies.

Learning Outcome (LO):

By the end of the course, students will be able to:

- Design and develop functional Android applications from scratch.
- Implement key Android features such as activities, services, and content providers.
- Create intuitive user interfaces using XML layouts and UI components.
- Perform network communication and data persistence in Android apps.
- Integrate multimedia elements and utilize device sensors.
- Employ advanced Android techniques such as notifications, maps, and external API integration.
- Apply the knowledge gained from case studies to build complex Android applications.

CourseOutline:

Unit1: Introductionto Android Development-Overview of Android ecosystem, setting up development environment (Android Studio, SDK, emulators), Understanding the Android project structure, Introduction to Android components (activities, fragments, services). User Interface Development - Layouts and views, User input and event handling, working with menus and dialogues, Creating responsive and adaptive interfaces, Material Design principles and guidelines

Unit2: Data Storage and Persistence-Using SQLite data base, Content Providers and data sharing Shared Preferences for app preferences, Working with files and external storage, Introduction to cloud storage and synchronization (Firebase). Networking and Web Services - Making HTTP requests (HTTP libraries, REST ful APIs), Parsing JSON and XML data, Working with web sockets, Authentication and authorization mechanisms, Caching and offline capabilities

Unit 3: Multimedia Integration- Workingwith images, audio, andvideo, integrating camera and gallery function alities, playing media files and streaming, Implementing notifications and push notifications, Location-basedservicesandmapsintegration,Testing and debugging Android applications. Case Studies and Project

Development-Exploring casestudies of popular Android applications (Data Collected),Analysing and dissecting their architecture andkey features (Social Media Applications), Implementing key components of case study apps, Applying best practices and optimization techniques,

Recommended Reference Books:

1. "Android Programming: The Big Nerd Ranch Guide"by Bill Phillips, Chris Stewart, and Kristin Marsicano
2. "Head First Android Development"by Dawn Griffiths and David Griffiths
3. "Android Studio 4.0 Development Essentials-Kotlin Edition"by Neil Smyth
4. "Android App Development for Dummies"by Michael Burton
5. "Professional Android 4 Application Development"by Reto Meier

Programming Group: Open Group

Course Title	Python		
Course Code (PR)	CIT/MJP /504	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisites:

- There are no specific prerequisites for this course. However, a basic understanding of programming concepts and familiarity with any programming language would be beneficial.

Course Objectives (CO):

- To introduce students to the fundamentals of Python programming language.
- To develop students' problem-solving skills using Python.
- To enable students to apply Python programming techniques to solve real-world problems.
- To provide hands-on experience in Python programming through practical exercises and case studies.
- To prepare students for further studies or career in software development, data analysis, or scientific computing.

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Understand and apply the core concepts of Python programming.
- Write Python programs to solve a variety of computational problems.
- Design and implement object-oriented programs in Python.
- Utilize Python libraries and modules for specific tasks.
- Analyze and debug Python code for errors and exceptions.
- Develop a basic understanding of GUI programming using Tkinter.

Course Outline:

Unit 1: Introduction to Python-History and features of Python, Installing Python and setting up the development environment, Basic Python syntax and data types, Variables, operators, and expressions. Control Structures, Conditional statements (if, else, elif), Looping statements (for, while), Control flow and branching

Unit 2: Functions and Modules-Defining and calling functions, Function parameters and return values, Recursion, Introduction to modules and libraries, Data Structures-Lists, tuples, and dictionaries, String manipulation, Sets and frozensets, Working with files and directories

Unit 3: Regular expressions, File handling and input/output operations, Debugging and error handling Introduction to GUI programming with Tkinter

Case Studies: Through out the course, students will work on case studies that demonstrate the practical application of Python programming in various domains, such as: a) Analyse and visualizing data using Python libraries like NumPy, Pandas, and Matplotlib, b) Building web applications with frameworks like Django or Flask., c) Implementing algorithms for machine learning and data mining, d) Automating tasks and working with APIs., e) Creating graphical user interfaces (GUI) for desktop applications.

Recommended Reference Book:

- "Python Programming: An Introduction to Computer Science", John Zelle, Franklin, Beedle & Associates Inc, 2016
- "Fluent Python" by Luciano Ramalho
- "Python Cook book" by David Beazley and Brian K. Jones
- "Python Crash Course" by Eric Matthes
- "Python for Data Analysis" by Wes

Course Title	Advance Python		
CourseCode(PR)	CIT/MJP /554	CourseType	Skill/Advance
Credits	2 Credits(PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisites:

- Proficiency in Python programming language, including knowledge of basic syntax, datatypes, control structures, and functions.
- Familiarity with object-oriented programming concepts.

Course Objectives(CO):

- Develop a deep understanding of advanced Python concepts and techniques.
- Master the design and implementation of complex data structures and algorithms in Python.
- Gain proficiency in object-oriented programming and apply design patterns effectively.
- Acquire knowledge of functional programming paradigms and utilize the minPython.
- Learn how to write Pythonic code and follow best practices for code organization and style.
- Understand concurrency and parallelism in Python and apply the mt to optimize performance.
- Enhance debugging and testing skills to ensure the equality of Python applications.

Learning Outcomes(LO):

By the end of this course, participants will be able to:

- Design and implement advanced data structures and algorithms in Python.
- Utilize object-oriented programming principles to develop modular and reusable code.
- Apply functional programming concepts to write elegant and concise Python code.
- Implement concurrency and parallelism in Python to improve performance.
- Write clean, maintainable, and Pythonic code following best practices.
- Effectively debug and test Python applications for identifying and fixing issues.
- Apply the learned concepts to real-world case studies and solve complex problems.

Course Outline:

Unit1: Advanced Data Structures and Algorithms in Python-Advanced data structures: sets, dictionaries, heaps, and graphs, Algorithm design and analysis: recursion, sorting, searching, and dynamic programming, Time and space complexity analysis

Unit 2: Object-Oriented Programming (OOP) in Python - Inheritance, polymorphism, and encapsulation, Design patterns and their implementation in Python, Advanced OOP concepts: abstract classes, interfaces, and multiple inheritance, Functional Programming in Python - Higher-order functions, lambda expressions, and closures, Immutable data structures and pure functions, Functional programming techniques: map, filter, reduce, and recursion

Unit3: Concurrency and Parallelism in Python-Multithreading and multiprocessing, Synchronization and thread safety, Asynchronous programming with async/await. Python Database Libraries - Overview of popular Python database libraries (e.g., SQL Alchemy, psycopg2), Installation and setup of database libraries, connecting to databases using Python. Database Querying with Python - Executing SQL queries using Python, Fetching and manipulating query results, Parameterized queries and prepared statements, Object-Relational Mapping(ORM)-Introduction to ORM frameworks (e.g., SQL Alchemy), Mapping database tables to Python classes, Performing CRUD operations using ORM

Case Studies: Throughout the course, students will analyse and work on various case studies to apply the advanced Python concepts they have learned. The case studies will cover domains such as data analysis, web development, scientific computing, and machine learning. These real-world scenarios will provide practical experience and enable participants to tackle complex problems using advanced Python programming techniques.

Recommended Reference Book:

"Python Cookbook", David Beazley and Brian K. Jones, O'Reilly Media, 2018 "Fluent Python" by Luciano Ramalho

"Python Cookbook" by David Beazley and Brian K. Jones "Python Crash Course" by Eric Matthes

"Python for Data Analysis" by Wes

Course Title	Open Web Programming (PHP)		
Course Code (PR)	CIT/MJP /604	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisite:

- Basic understanding of programming concepts
- Familiarity with HTML, CSS, and Java Script is preferred but not mandatory

Course Objectives(CO):

- To introduce students to the fundamentals of PHP programming language
- To enable students to develop dynamic web applications using PHP
- To provide hands-on experience with data base integration and websecurity in PHP
- To enhance students' problem-solving and critical thinking skills in the context of PHP development

Learning Outcome(LO):

By the end of this course, students will be able to:

- Write PHP code to solve programming problems and develop web applications
- Integrate PHP with data bases for efficient data management
- Implement security measures to protect web applications from common vulnerabilities
- Optimize PHP code and data base queries for improved performance
- Analyze and trouble shoot PHP applications using debugging tools and techniques

Course Outline:

Unit1: Introduction to PHP-Basics of PHP syntax, Variables and datatypes, Control structures (if-else, loops), Functions and array, PHP Programming Fundamentals-File handling and input/output operations, String manipulation, Regular expressions, Error handling and debugging

Unit2: Object-Oriented Programming in PHP-Classes and objects, Inheritance and polymorphism, Encapsulation and data abstraction, Exception handling. Data base Integration-Introduction to data bases (MySQL, SQLite, etc.), SQL queries and data base operations, Connecting PHP with data bases, CRUD operations (Create, Read, Update, Delete).

Unit 3. Web Application Development - Basics of web development (HTML, CSS, Java Script), Server-side scripting and client-server communication, Handling form (GET, POST) submissions, Session management and cookies. Security Considerations - Common web vulnerabilities (cross-site scripting, SQL injection), Input validation and data sanitization, Password hashing and encryption, User authentication and authorization. Performance Optimization - Caching techniques, Code profiling and optimization, Database query optimization, Load balancing and scale ability considerations

Recommended Reference Book:

"PHP and MySQL Web Development" by Luke Welling and Laura Thomson.

Programming Group: Web Scripting Group

Course Title	VB & Java Script		
Course Code (PR)	CIT/MJP /505	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisite:

No prior programming experience is required for this course. However, a basic understanding of computer concepts and familiarity with using computers and the internet is recommended.

Course Objectives (CO):

- To introduce students to the fundamentals of VB and Java Script programming languages.
- To enable students to create interactive web pages and develop desktop applications using VB.
- To familiarize students with the integration of VB and Java Script for enhanced functionality.
- To develop problem-solving skills through hands-on case studies and practical assignments.
- To prepare students for further studies or careers in web development and software engineering.

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Understand and apply the syntax, concepts, and principles of VB and Java Script programming.
- Develop web pages with interactive elements and dynamic content using Java Script.
- Design and implement desktop applications using VB, incorporating Java Script functionality.
- Solve programming problems using logical thinking and debugging techniques.
- Apply VB and Java Script programming knowledge to real-world scenarios through case studies.

Course Outline:

Unit1: Introduction to Visual Basic Programming-Overview of Visual Basic programming language, Introduction to the Integrated Development Environment (IDE), Basic syntax and data types in Visual Basic, Variables, operators, and expressions, Control structures: decision-making and looping, Arrays and collections. Advanced Visual Basic Programming - Object-oriented programming concepts in Visual Basic, Classes, objects, and inheritance, Exception handling and error trapping, File handling and data input/output, User interface design with forms and controls, Event-driven programming

Unit 2: Introduction to JavaScript Programming - Introduction to JavaScript and its role in web development, JavaScript syntax, variables, and data types, Control flow and conditional statements, Functions and scope, Working with arrays and objects, DOM manipulation and event handling. Advanced JavaScript Programming-Advanced JavaScript concepts: closures, prototypes, and modules, Asynchronous programming with JavaScript, Error handling and debugging techniques, working with JSON and AJAX, Introduction to modern JavaScript frameworks (e.g., React, Angular)

Unit 3: Case Studies: a) Building a Desktop Application with Visual Basic: Students will develop a desktop application using Visual Basic, incorporating various concepts covered in the course. The case study will focus on user interface design, data management, and implementing business logic b) Creating Dynamic Web Pages with JavaScript: Students will create interactive web pages using JavaScript. They will learn to manipulate the

Document Object Model (DOM), handle events, and retrieve data from external sources, showcasing the power of JavaScript in web development.

Recommended Reference Books:

"Visual Basic 2019 in 24 Hours, Sams Teach Yourself" by James Foxall "Murach's JavaScript and jQuery" by Zak Ruvalcaba and Mary Delamater

"Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke "JavaScript: The Good Parts" by Douglas Crockford

"Professional Visual Basic 2019 and .NET Core 3.0" by Bill Sheldon, Billy Hollis, and Rob Windsor

Course Title	Node.js		
Course Code (PR)	CIT/MJP /555	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisite:

- Basic knowledge of JavaScript programming language
- Familiarity with web development concepts (HTML, CSS, and client-side JavaScript)
- Understanding of fundamental concepts of server-client architecture

Course Objectives (CO):

- Gain a thorough understanding of Node.js and its key features
- Develop skills to build server-side applications using Node.js
- Learn best practices for a synchronous programming and handling I/O operations
- Acquire knowledge of web development with Node.js, including routing, middleware, and database integration
- Explore advanced topics such as real-time communication, microservices, and performance optimization
- Analyze and comprehend real-world case studies to apply Node.js effectively in various scenarios

Learning Outcomes (LO):

Upon successful completion of the course, students will be able to:

- Build robust web applications using Node.js and related frameworks
- Implement asynchronous programming techniques to handle concurrent operations efficiently
- Integrate databases and implement user authentication in Node.js applications
- Apply best practices for testing, debugging, and deployment of Node.js applications
- Analyze and understand the architecture and implementation of real-world Node.js projects
- Solve complex programming challenges using the knowledge gained during the course

Course Outline:

Unit 1: Introduction to Node.js - Overview of Node.js and its features, Understanding the event-driven, non-blocking I/O model, Installing Node.js and setting up the development environment, Introduction to npm (NodePackage Manager). JavaScript Fundamentals - Revision of JavaScript essentials, Asynchronous programming concepts and callbacks, Promises and async/await for handling asynchronous operations

Unit 2: Building Web Servers with Node.js - Creating a basic HTTP server using the built-in HTTP module, Routing and handling different types of requests, Working with Express.js, a popular Node.js web application framework. Working with Databases - Overview of data base systems and their role in web applications, Introduction to MongoDB and NoSQL databases, Using MongoDB with Node.js and Mongoose ODM (Object-Data Mapping)

Unit 3: Asynchronous Programming in Node.js - Understanding the Event Loop and non-blocking I/O, Using callbacks, Promises, and async/await for asynchronous programming, Error handling and best practices, Middleware and Authentication - Implementing middleware for request processing, User authentication and authorization techniques, Working with JSON Web Tokens (JWT) for secure authentication. Real-time Applications with Socket.io, Introduction to real-time communication, Building real-time web applications with Socket.io, Broad casting and handling events in real-time Case Studies a) Analyzing and implementing real-world Node.js applications, b) Examining scalability, performance optimization, and best practices, Case studies may include echat applications, API servers, and more

Recommended Reference Books:

"Node.js Web Development" by David Herron

"Learning Node.js: A Hands-On Guide to Building Web Applications in Java Script "by Marc Wandschneider" Node.js Design

Patterns" by Mario Casciaro and Luciano Mammino

"Mastering Node.js "by Sandro Pasquali "Node.js in Action" by Mike Cantelon, Marc Harter, T. J. Holowaychuk, and Nathan Rajlich

Course Title	React		
Course Code (PR)	CIT/MJP /605	Course Type	Skill/Advance
Credits	2 Credits (PR)	Contact Hours	4 Hrs / Week
Level	To be conducted in Programming-1 Elective Basket of Programming Group at Semester 1		

Prerequisites:

- Basic knowledge of HTML, CSS, and Java Script
- Familiarity with web development concepts and principles

Course Objectives (CO):

- Gain thorough understanding of React and its core concepts
- Learn how to build interactive and dynamic user interfaces using React components
- Develop proficiency in managing state and handling events in React
- Understand the fundamental of React Router for building single-page applications

- Acquire knowledge of popular styling approaches and libraries for React
- Learn form handling and validation techniques in React
- Explore advanced concepts such as React hooks and performance optimization

Learning Outcomes(LO):

By the end of this course, students will be able to:

- Build responsive and interactive web applications using React
- Develop reusable and modular React components
- Implement efficient state management and event handling in React applications
- Create single-page applications with React Router
- Apply styling techniques to enhance the visual appeal of React applications
- Implement form handling and validation in React
- Understand and apply advanced React concepts for optimal performance and error handling

Course Outline:

Unit 1: Introduction to React-

Overview of React and its key features, Understanding the React component model, Setting up a development environment for React, Introduction to JSX (JavaScript XML), React Components and State Management - Creating functional and class components, understanding component lifecycle methods, managing component state with hooks and context, Handling events and data binding in React

Unit 2: React Routing and Navigation- Introduction to React Router for handling client-

side routing, Implementing navigation and nested routes in React, Managing route parameters and query strings, Implementing dynamic routing and code splitting, Styling in React - Styling options in React: CSS, inline styles, CSS modules, Using popular CSS-in-JS libraries (e.g., styled-components), Best practices for organizing and managing styles in React

Unit 3: Advanced React Concepts- State management with Redux or MobX, integrating third-party libraries and APIs with React, Server-side rendering with React (Next.js), Testing and debugging React applications, React Forms and Validation: Controlled and uncontrolled components, Form handling and validation techniques, Form libraries and formik integration, Advanced React Concepts: React hooks for custom logic (useReducer, useContext, etc.), Performance optimization techniques, Error handling and debugging in React

Recommended Reference Books:

"Learning React: Modern Patterns for Developing React Apps" by Alex Banks and Eve Porcello "ReactUp and

Running: Building Web Applications" by Stoyan Stefanov

"Pro React 16" by Adam Freeman

"Fullstack React: The Complete Guide to ReactJS and Friends" by Anthony Accomazzo, Ari Lerner, David Guttman, and Nate Murray

"React Design Patterns and Best Practices: Build easy to scale modular applications using the most powerful components and design patterns" by Carlos Santana Roldán

Discipline Specific Elective Basket

Course Contents for Discipline Specific Electives – DSE (Elective Group Basket)

Elective Group	Elective 1 CIT/DSET/520-522 CIT/DSEP/520-522	Elective 2 CIT/DSET/570-572 CIT/DSEP/570-572	Elective 3 CIT/DSET/620-622 CIT/DSEP/620-622	Elective 4 CIT/DSET/670-672 CIT/DSEP/670-672
Pattern Analysis & Machine Intelligence	Soft Computing	Fuzzy Systems : Theory, Application & Case Study	Video Processing	Pattern Recognition
Security	Network Security	Cyber Security	Cyber Forensics: Tools, Techniques and Case Studies	Cryptography & Blockchain
Natural Language Processing	Linguistic Fundamentals: Understanding Language Structure and Analysis	Semantics and Pragmatics	Natural Language Processing	AI Chatbot Services and Applications

Note: * Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Elective Group – Pattern Analysis and Machine Intelligence

Course Title	Soft Computing		
Course Code (TH)	CIT/DSET/ 520	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/520		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of mathematics and probability theory
- Understanding of computer programming concepts
- Familiarity with algorithms and data structures
- Basic understanding of artificial intelligence and machine learning

Course Objectives (CO):

- To introduce students to the fundamental concepts and techniques of soft computing
- To provide a comprehensive understanding of fuzzy logic systems, neural networks, and evolutionary computation
- To explore the integration of different soft computing methodologies
- To equip students with the skills to apply soft computing techniques to real-world problems
- To analyze and implement case studies that demonstrate the effectiveness of soft computing approaches

Learning Outcome (LO):

By the end of this course, students will be able to:

- Understand the principles and theories underlying soft computing techniques
- Apply fuzzy logic systems, neural networks, and evolutionary computation to solve complex problems
- Analyze and evaluate the performance of soft computing models
- Design and develop soft computing-based solutions for real-world case studies

Unit 1: Introduction to Soft Computing - Introduction to soft computing and its significance, Comparison with traditional computing paradigms, Components of soft computing: Neural networks, fuzzy logic, and evolutionary computation, Soft computing in real-world applications. **Neural Networks** - Introduction to artificial neural networks (ANNs), Single-layer perceptron and multi-layer perceptron, Training algorithms: Back propagation, gradient descent, and variants, Deep learning and convolutional neural networks (CNNs), Case study: Image recognition using ANNs

Unit 2: Fuzzy Logic - Introduction to fuzzy logic and fuzzy sets, Fuzzy rules and linguistic variables, Fuzzy inference systems and rule-based reasoning, Fuzzy control systems and applications Case study: Fuzzy logic-

based temperature control system. **Evolutionary Computation** - Introduction to evolutionary computation and genetic algorithms, Representation schemes and fitness evaluation, Selection, crossover, and mutation operators, Genetic programming and applications Case study: Optimization using genetic algorithms

Unit 3: Hybrid Approaches and Applications - Hybridization of soft computing techniques, Neuro-fuzzy systems and their applications, Evolutionary neural networks, Soft computing in data mining, pattern recognition, and decision support systems, Case study: Hybrid approach for stock market prediction

Recommended Reference Books:

"Soft Computing: Techniques and Applications" by S.N. Sivanandam and S.N. Deepa
"Neural Networks and Learning Machines" by Simon Haykin

"Fuzzy Logic with Engineering Applications" by Timothy J. Ross

"Introduction to Evolutionary Computing" by A. E. Eiben and J.E. Smith

"Hybrid Intelligent Systems: Analysis and Design" by Siddhartha Bhattacharyya and Paramartha Dutta

Course Title	Fuzzy Systems : Theory, Application and Case Studies		
Course Code (TH)	CIT/DSET/ 570	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/570		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of mathematics, including set theory and calculus.
- Familiarity with basic concepts of logic and reasoning.
- Basic programming skills (preferably in a language such as Python or MATLAB).

Course Objectives (CO):

- Understand the theoretical foundations of fuzzy systems and their applications.
- Gain knowledge of fuzzy logic, fuzzy sets, and fuzzy inference systems.
- Learn techniques for fuzzy system modeling, control, and decision-making.
- Develop the ability to design and implement fuzzy systems for real-world problems.
- Analyze and evaluate case studies to understand the practical implications of fuzzy systems.

Learning Outcomes (LO):

By the end of the course, students will be able to:

- Explain the principles and concepts underlying fuzzy systems.
- Design and implement fuzzy inference systems for various applications.
- Apply fuzzy modeling techniques to develop solutions for complex problems.
- Evaluate and analyze the performance of fuzzy systems in real-world scenarios.
- Apply fuzzy decision-making methods to support complex decision processes.

Course Outline:

Unit 1: Introduction to Fuzzy Systems-

Introduction to uncertainty and imprecision, Fuzzy sets and membership functions, Fuzzy operations and linguistic variables, Fuzzy rules and fuzzy reasoning. Fuzzy Systems Design and Implementation - Fuzzy inference systems, Rule-based fuzzy systems, Fuzzy control systems, Fuzzy decision-making

Unit 2: Fuzzy Systems in Engineering Applications - Fuzzy control systems in robotics, Fuzzy modelling and control in process industries, Fuzzy systems in intelligent transportation systems, Fuzzy systems for pattern recognition. Fuzzy Systems in Finance and Economics - Fuzzy logic in portfolio optimization, Fuzzy modeling for risk assessment, Fuzzy time series forecasting, Fuzzy systems in credit scoring

Unit 3: Fuzzy Systems in Medical and Healthcare Applications - Fuzzy expert systems for medical diagnosis, Fuzzy systems for healthcare resource allocation, Fuzzy decision support systems in medical treatment, Fuzzymodelingofpatientsatisfaction, CaseStudiesa) Analyzingandimplementingfuzzysystemsusingsoftwaretools, b) Casestudiesfromvariousdomains(engineering,finance,medicine),c)Evaluationandperformanceassessmentof fuzzysystems

Recommended Reference Books:

"Fuzzy Logic with Engineering Applications" by Timothy J. Ross

"Fuzzy Sets and Fuzzy Logic: Theory and Applications" by George J. Klir and Bo Yuan

"Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems" by Guanrong Chen and Trung Tat Pham

"Fuzzy Systems Engineering: Theory and Practice" by C.L. Philip Chen and Han-Pang Huang

"Fuzzy Logic: Intelligence, Control, and Information" by John Yen and Reza Langari

Course Title	Video Processing		
Course Code (TH)	CIT/DSET/ 620	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/620		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Proficiency in Python programming language
- Basic understanding of image processing concepts
- Familiarity with fundamental concepts of computer vision

Course Objectives (CO):

- Understand the fundamental principles and techniques of video processing.
- Gain proficiency in using Python libraries and frameworks for video manipulation and analysis.
- Apply various video processing techniques to extract meaningful information from video streams.
- Acquire the skills to enhance and restore video quality using advanced algorithms.
- Explore the application of deep learning for video analysis and synthesis.
- Develop the ability to design and implement video processing solutions for real-world problems.

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Understand the underlying concepts and techniques of video processing.
- Manipulate and preprocess video data using Python.
- Implement motion detection, object recognition, and face detection algorithms.
- Enhance video quality through denoising, deblurring, and color correction.
- Apply video compression techniques for efficient storage and transmission.
- Utilize deep learning models for video analysis, classification, and synthesis.
- Design and develop video processing solutions for real-world applications.

Course Outline:

Unit 1: Introduction to Video Processing - Fundamentals of video representation and formats, Video acquisition and preprocessing techniques. Video Analysis Techniques - Motion detection and tracking, Object recognition and tracking, Face detection and recognition

Unit 2: Video Enhancement and Restoration - Noise reduction and image stabilization, Video denoising and deblurring, Contrast enhancement and color correction. Video Compression and Encoding, Principles of video compression, Video encoding algorithms and standards, Codecs and formats for efficient storage and transmission

Unit 3: Deep Learning for Video Processing - Convolutional neural networks for video analysis, Video classification and action recognition, Video generation and synthesis. Case Studies a) Real-world applications of video processing using Python, b) Case studies on surveillance systems, video analytics, etc. c) Hands-on projects to reinforce concepts learned.

Recommended Reference Book:

"Python for Video Processing: Techniques and Case Studies"

"Digital Video Processing", Second Edition by A. Murat Tekalp, June 2015, Pearson, ISBN: 9780133991116

Course Title	Pattern Recognition		
Course Code (TH)	CIT/DSET/ 670	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/670		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of Python programming language
- Familiarity with fundamental concepts in machine learning
- Understanding of linear algebra and probability theory

Course Objectives (CO):

- To understand the fundamental principles and techniques of pattern classification
- To develop proficiency in implementing pattern classification algorithms using Python
- To gain practical experience through hands-on exercises and real-world case studies
- To learn how to evaluate and select appropriate classification models for different tasks
- To explore the ethical considerations and challenges in pattern classification

Learning Outcome (LO):

By the end of this course, students will be able to:

- Understand the concepts and principles of pattern classification
- Apply various pattern classification algorithms using Python
- Preprocess and analyze datasets for pattern classification tasks
- Evaluate and compare the performance of classification models
- Implement pattern classification solutions to real-world problems

Course Outline:

Unit 1: Introduction to Pattern Classification - Overview of pattern classification, Importance and applications of pattern classification, Key concepts and terminology. Data Preprocessing and Feature Selection - Data cleaning and preprocessing techniques, Feature extraction and dimensionality reduction methods, Feature selection algorithms

Unit 2: Supervised Learning Algorithms - Linear classifiers (e.g., logistic regression, support vector machines), Decision trees and ensemble methods (e.g., random forests, boosting), Neural networks and deep learning models. Unsupervised Learning Algorithms - Clustering techniques (e.g., k-means, hierarchical clustering), Density estimation and Gaussian mixture models, Dimensionality reduction methods (e.g., principal component analysis)

Unit 3: Evaluation Metrics and Model Selection - Performance evaluation measures (e.g., accuracy, precision, recall), Cross-validation techniques, Model selection and hyperparameter tuning. Pattern Classification with Python - Overview of Python libraries for pattern classification (e.g., scikit-learn, TensorFlow, Keras), Implementing classification algorithms using Python, Handling large datasets and optimizing performance.

Recommended Reference Book:

"Pattern Classification and Machine Learning", Christopher M. Bishop, Springer, 2006 "Pattern Classification", Richard O. Duda, Peter E. Hart, David G. Stork, Wiley

Elective Group–Security

Course Title	Network Security		
Course Code (TH)	CIT/DSET/ 521	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/521		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of networking concepts
- Familiarity with operating systems and computer architecture
- Understanding of fundamental security principles

Course Objectives (CO):

- Understand the fundamental principles and concepts of network security.
- Identify and assess network security risks and vulnerabilities.
- Evaluate and implement appropriate network security technologies and protocols.
- Develop incident response plans and perform network forensics.
- Analyze and apply security measures in cloud and mobile network environments.
- Analyze and learn from real-world network security case studies.

Learning Outcomes (LO):

By the end of this course, students will be able to:

- Apply network security principles to design secure network infrastructures.
- Implement and configure network security technologies effectively.
- Analyze and respond to network security incidents.
- Evaluate and apply security measures in cloud and mobile networks.
- Investigate and present case studies of network security breaches.

Course Outline:

Unit 1: Introduction to Network Security -

Overview of network security concepts, Security goals: confidentiality, integrity, availability, Threat landscape and attack vectors, Risk assessment and management, Network Architecture and Design - Secure network design principles, Segmentation and zoning, Defense-in-depth strategies, Security considerations for wireless and mobile networks

Unit 2: Network Perimeter Security - Firewalls and intrusion detection/prevention systems, Virtual Private Networks (VPNs), Network Address Translation (NAT), Demilitarized Zone (DMZ) design, Secure Network Protocols and Services - Secure Socket Layer/Transport Layer Security (SSL/TLS), Secure Shell (SSH), Domain Name System Security Extensions (DNSSEC), Network Time Protocol Security (NTPsec)

Unit 3: Intrusion Detection and Prevention Systems (IDPS) - Host-based and network-based IDPS, Signature-based and anomaly-based detection, Incident response and handling. Wireless Network Security - Wi-Fi security protocols (WEP, WPA, WPA2, WPA3), Rogue access point detection, Wireless intrusion prevention systems (WIPS), Securing mobile devices and applications. Case Studies in Network Security a) Analysis of real-world security breaches b) Investigating network attacks and incidents c) Lessons learned from notable security incidents

Recommended Reference Books:

"Network Security: Private Communication in a Public World" by Charlie Kaufman, Radia Perlman, and Mike Speciner

"Firewalls and Internet Security: Repelling the Wily Hacker" by William R. Cheswick and Steven M. Bellovin
"Network Security Essentials: Applications and Standards" by William Stallings

"Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems" by Chris Sanders

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto

"Hacking: The Art of Exploitation" by Jon Erickson

"Applied Cryptography: Protocols, Algorithms, and Source Code in C" by Bruce Schneier

"Security Engineering: A Guide to Building Dependable Distributed Systems" by Ross J. Anderson

Course Title	Cyber Security		
Course Code (TH)	CIT/DSET/ 571	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/571		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisite:

Basic understanding of computer networks and information technology.

Course Objectives (CO):

- To provide students with a solid foundation in cyber security concepts and principles.
- To develop an understanding of common cyber threats and vulnerabilities.
- To familiarize students with various tools and techniques used in cyber security.
- To enable students to assess and mitigate risk to digital systems and data.
- To promote ethical considerations and legal compliance in cyber security practices.

Learning Outcomes(LO):

By the end of the course, students will be able to:

- Identify and analyze common cyber threats and vulnerabilities.
- Implement network security measures to protect against attacks.
- Apply cryptographic techniques to ensure data confidentiality and integrity.
- Develop and implement secure software development practices.
- Demonstrate incident response and management skills.
- Conduct ethical hacking and penetration testing activities.
- Evaluate and recommend security measures for different scenarios.

Course Outline:

Unit 1: Introduction to Cyber Security - Overview of cyber security fundamentals, Importance of cyber security in modern society, Legal and ethical considerations, Cyber Threats and Attacks - Types of cyber threats: malware, social engineering, phishing, etc. Attack vectors and methods, Risk assessment and vulnerability analysis

Unit 2: Cybersecurity Tools - Firewall and intrusion detection systems, Anti-malware software and endpoint protection, Network monitoring and log analysis tools, Encryption and data protection tools, Securing Networks and Systems - Network security principles and protocols, Secure configuration of operating systems and applications, Authentication and access control mechanisms, Patch management and vulnerability mitigation

Unit 3: Incident Response and Forensics - Incident response lifecycle, Digital forensics techniques, Evidence collection and analysis, Post-incident recovery and lessons learned, Case Studies in Cyber Security – a) Analysis of real-world cybersecurity incidents, b) Examination of incident response strategies c) Lessons learned and best practices

Recommended Reference Books:

"The Art of Invisibility" by Kevin Mitnick

"Hacking: The Art of Exploitation" by Jon Erickson

"Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" by Michael Sikorski and Andrew Honig

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto

"Network Security Bible" by Eric Cole

"Blue Team Handbook: Incident Response Edition" by Don Murdoch, Jr.

"Digital Forensics and Incident Response: Developing an Incident Response Plan" by Gerard Johansen

Course Title	Cyber Forensics: Tools, Techniques and Case Studies		
Course Code (TH)	CIT/DSET/ 621	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/621		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of computer systems and networks
- Familiarity with operating systems and file systems
- Understanding of information security principles

Course Objectives(CO):

- Understand the fundamentals of cyber forensics and its role in combating cybercrime.
- Gain proficiency in the use of tools and techniques for acquiring and analyzing digital evidence.
- Develop skills in conducting forensic investigations and incident response procedures.
- Comprehend the legal and ethical issues associated with cyber forensics.
- Explore advanced topics such as network forensics, mobile and cloud forensics, and malware analysis.

Learning Outcome(LO):

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

- Identify and collect digital evidence using forensically sound methods.
- Analyze digital artifacts to uncover evidence of cybercrime.
- Conduct network and mobile device forensics investigations.
- Employ appropriate tools and techniques to recover and preserve data.
- Understand the legal and ethical considerations in cyber forensics.

Course Outline:

Unit 1: Introduction to Cyber Forensics - Understanding cyber forensics: scope and importance, Legal and ethical considerations in cyber investigations, Roles and responsibilities of a cyber forensic examiner. Digital Evidence and Forensic Processes, Types of digital evidence, Evidence acquisition and preservation, Forensic imaging and hashing, Chain of custody and documentation

Unit 2: Forensic Tools and Techniques - Forensic imaging tools (e.g., EnCase, FTK, Autopsy), File system analysis and recovery, Network forensics and log analysis, Memory forensics and analysis, Mobile and Cloud

Forensics, Forensics for mobile devices (smartphones, tablets), Investigating cloud-based platforms and services, Extraction and analysis of mobile app data

Unit 3: Network Traffic Analysis - Packet capture and analysis, Intrusion detection and prevention systems, Network log analysis, Malware Analysis and Reverse Engineering - Introduction to malware analysis, Static and dynamic analysis techniques, Code decompilation and reverse engineering. Case Studies and Practical Application

– a) Analyzing a cybercrime case from start to finish, b) Examining digital evidence and conducting forensic analysis, c) Reporting findings and presenting evidence in court

Recommended Reference Books:

"Digital Forensics and Cyber Crime: An Introduction" by Marjorie T.

Britz "Computer Forensics: Investigating Data and Image Files" by EC-Council

"The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics" by John Sammons "File System Forensic Analysis" by Brian Carrier

"Malware Forensics: Investigating and Analyzing Malicious Code" by Cameron H. Malin, et al.

"Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems" by Chris Sanders

Course Title	Cryptography and Blockchain		
Course Code (TH)	CIT/DSET/ 671	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/671		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisite:

- Basic understanding of computer science concepts and programming
- Familiarity with data structures and algorithms
- Knowledge of networking and internet protocols

Course Objectives (CO):

- Understand the fundamental concepts and principles of cryptography and blockchain technology.
- Analyze and evaluate different cryptographic algorithms, protocols, and systems.
- Gain hands-on experience in implementing and securing cryptographic systems.
- Explore the inner workings of blockchain networks and consensus mechanisms.
- Develop the skills to design and deploy decentralized applications using smart contracts.
- Investigate real-world use cases of cryptography and blockchain in various industries.

Learning Outcome (LO):

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

- Demonstrate a comprehensive understanding of cryptographic algorithms and their applications.
- Evaluate the security of cryptographic systems and propose countermeasures.
- Analyze and design blockchain networks for specific use cases.
- Develop and deploy smart contracts for decentralized applications.
- Apply cryptography and blockchain principles to address real-world challenges.

Course Outline:

Unit 1: Introduction to Cryptography - Overview of cryptography and its historical significance, Symmetric and asymmetric encryption algorithms, Hash functions and digital signatures, Cryptographic protocols and applications. Cryptographic Techniques and Algorithms - Key management and distribution, Public key infrastructure (PKI), Cryptographic attacks and countermeasures, Secure communication protocols (SSL/TLS)

Unit 2: Introduction to Blockchain Technology -

Evolution of blockchain and its applications, Distributed ledger technology and consensus mechanisms, Cryptocurrencies and smart contracts, Blockchain platforms and ecosystems, Blockchain Security and Privacy - Blockchain vulnerabilities and attack vectors, Privacy and anonymity in blockchain, Tokenization and non-fungible tokens (NFTs), Securing blockchain networks and transactions

Unit 3: Future Trends and Challenges - Quantum cryptography and post-quantum encryption, Scalability and interoperability in blockchain, Regulatory and legal considerations, Ethical implications of cryptography and blockchain technology, Case Studies in Cryptography and Blockchain - a) Real-world applications of cryptography in finance, healthcare, and government sectors, b) Success stories and challenges of blockchain adoption in industries such as supply chain, voting, and identity management.

Recommended Reference Books:

"Cryptography and Network Security: Principles and Practice" by William

Stallings "Mastering Bitcoin: Unlocking Digital Cryptocurrencies" by Andreas M. Antonopoulos "Blockchain Basics: A Non-Technical Introduction in 25 Steps" by Daniel Drescher

"Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World" by Don Tapscott and Alex Tapscott

"The Age of Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" by Paul Vigna and Michael J. Casey

"Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" by Abraham K. White

Elective Group–Natural Language Processing

CourseTitle	Linguistic Fundamentals: Understanding Language Structure and Analysis		
CourseCode(TH)	CIT/DSET/ 522	Course Type	Mandatory
CourseCode(PR)	CIT/DSEP/522		
Credits	4 Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisite:

None. This course is open to all students with an interest in linguistics. Prior knowledge of linguistics is not required.

Course Objectives(CO):

- To develop a comprehensive understanding of the fundamental components of language.
- To examine the interaction between linguistic elements and their impact on language structure.
- To cultivate analytical and critical thinking skills in linguistic analysis.
- To explore language variation and its social and cultural implications.
- To apply linguistic theories and methods to real-world case studies.

Learning Outcomes(LO):

By the end of this course, students will be able to:

- Identify and describe the core elements of language, including phonetics, phonology, morphology, syntax, semantics, and pragmatics.
- Analyze and transcribe speech sounds using phonetic symbols.
- Conduct morphological and syntactic analyses of linguistic data.
- Interpret and analyze meaning at the lexical, sentence, and discourse levels.
- Apply sociolinguistic principles to examine language variation and change.
- Apply linguistic theories and methods to case studies, demonstrating critical thinking skills.

Course Outline:

Unit 1: Introduction to Linguistic Fundamentals - Overview of linguistics as a field of study, Language as a system of communication, Fundamental branches of linguistics. Phonetics and Phonology - Introduction to phonetics: articulatory, acoustic, and auditory aspects, Phonemic analysis: phonemes, allophones, and phonological rules, Case study: Phonological variations across different dialects

Unit 2: Morphology - Basic units of meaning: morphemes, Word formation processes: affixation, compounding, derivation, etc. Case study: Analyzing morphological structures in different languages. Syntax - Sentence structure and phrase types, Syntactic categories and constituents, Case study: Parsing and analyzing sentence structures

Unit 3: Semantics and Pragmatics - Meaning and interpretation of words, phrases, and sentences, Pragmatic principles and implicatures, Case study: Pragmatic analysis of speech acts, Language and Society - Sociolinguistics: language variation and social factors, Language acquisition and bilingualism Case study: Language policy and planning

Recommended Reference Books:

"An Introduction to Language" by Victoria Fromkin, Robert Rodman, and Nina Hyams

"Linguistics: An Introduction to Linguistic Theory" by Victoria A. Fromkin, Robert Rodman, and Nina

Hyams "The Cambridge Encyclopedia of Language" by David Crystal

"Language Files: Materials for an Introduction to Language and Linguistics" by Department of Linguistics, Ohio State University

"Introducing Morphology" by Rochelle Lieber

Course Title	Semantics and Pragmatics		
Course Code (TH)	CIT/DSET/ 572	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/572		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of linguistics or a related field is recommended but not required.
- Familiarity with introductory-level syntax and morphology is beneficial.

Course Objectives (CO):

- Understand the key concepts and theories of semantics and pragmatics.
- Develop analytical skills to identify and analyze different layers of meaning in language.
- Recognize and evaluate the role of context in shaping linguistic interpretation.
- Apply theoretical knowledge to real-world case studies and practical scenarios.
- Enhance critical thinking and problem-solving abilities within the field of linguistics.
- Develop effective communication skills through class discussions and presentations.

Learning Outcomes (LO):

By the end of the course, students will be able to:

- Demonstrate a comprehensive understanding of the core principles of semantics and pragmatics.
- Analyze and interpret the meaning of words, sentences, and discourse in different contexts.
- Evaluate and discuss the impact of pragmatic factors on language use and interpretation.
- Apply theoretical frameworks to analyze and solve problems related to semantics and pragmatics.
- Construct well-supported arguments and engage in academic discussions on language meaning.
- Employ critical thinking to assess and interpret linguistic phenomena in various domains.

Course Outline:

Unit 1: Introduction to Semantics and Pragmatics - Overview of semantics and pragmatics, Distinction between meaning and use, Theoretical frameworks in semantics and pragmatics. Semantic Analysis - Word meaning and lexical semantics, Sentence meaning and compositional semantics, Truth-conditional semantics

Unit 2: Pragmatic Analysis - Context and implicature, Speech acts and illocutionary force, Conversational implicature and inference, Reference and Presupposition - Reference and referring expressions, Presupposition and presuppositional analysis, Anaphora and deixis

Unit 3: Meaning and Society - Pragmatics of politeness and face-saving, Cross-cultural and intercultural pragmatics, Gender and language use, Case Studies and Applications a) Analyzing discourse and conversation b) Pragmatics in legal and political discourse, c) Semantic and pragmatic analysis in advertising

Recommended Reference Books:

"Semantics: A Coursebook" by James R. Hurford, Brendan Heasley, and Michael B. Smith "Pragmatics and Discourse: A Resource Book for Students" by Joan Cutting

"Meaning and Relevance" by Deirdre Wilson and Dan Sperber "Pragmatics" by Stephen C. Levinson

"Semantics: An Introduction to Meaning in Language" by Kate Kearns "Pragmatics: An Introduction" by Yan Huang

Course Title	Natural Language Processing		
Course Code (TH)	CIT/DSET/ 622	Course Type	Mandatory
Course Code (PR)	CIT/DSEP/622		
Credits	4 Credits (2TH+2PR)	Contact Hours (TH)	2 Hrs / Week
		Contact Hours (PR)	4 Hrs / Week

Prerequisites:

- Basic knowledge of programming concepts
- Familiarity with data structures and algorithms
- Understanding of probability and statistics
- Some exposure to machine learning concepts is beneficial but not mandatory

Course Objectives (CO):

- Understand the fundamental concepts and techniques in Natural Language Processing
- Develop practical skills to build and evaluate NLP systems
- Gain knowledge of various NLP applications and case studies
- Acquire the ability to analyze and solve NLP problems effectively
- Explore the challenges and ethical considerations in NLP

Learning Outcomes (LO):

By the end of the course, students should be able to:

- Explain the key concepts and methodologies in NLP
- Apply NLP techniques to preprocess and analyze text data
- Build and evaluate NLP models for tasks like sentiment analysis, named entity recognition, machine translation, and question answering
- Critically analyze and compare different approaches and algorithms in NLP
- Identify and address ethical concerns in NLP applications

Unit 1: Introduction to Natural Language Processing - Overview of NLP: Definition, history, and key applications. Language representation: Text preprocessing, tokenization, stemming, and lemmatization. Language modeling: N-grams, language models, and text generation. Evaluation metrics: Precision, recall, F1 score, and perplexity.

Unit 2: NLP Techniques and Algorithms - Text classification: Naive Bayes, logistic regression, and support vector machines (SVM). Sentiment analysis: Lexicon-based approaches, machine learning models, and deep learning models. Named Entity Recognition (NER): Rule-based systems, conditional random fields (CRF), and neural networks. Sequence labeling: Hidden Markov models (HMM) and Conditional Random Fields (CRF). Word embeddings: Word2Vec, GloVe, and FastText.

Unit 3: Advanced NLP - Topic modeling: Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). Text summarization: Extractive and abstractive approaches. Machine translation: Statistical methods, neural machine translation (NMT), and transformer models. Question answering: Information retrieval, document retrieval, and passage ranking. NLP in industry: Case studies of NLP applications in various domains, such as healthcare, finance, and customer support.

Recommended Reference Books:

"Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" by Daniel Jurafsky and James H. Martin.

"Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit" by Steven Bird, Ewan Klein, and Edward Loper.

"Foundations of Statistical Natural Language Processing" by Christopher D. Manning and Hinrich

Schütze. "Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain, and Karan Kumar.

"Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS" by Goutam Chakraborty and Murali Pagolu.

CourseTitle	AI Chatbot Services and Applications		
CourseCode(TH)	CIT/DSET/ 672	Course Type	Mandatory
CourseCode(PR)	CIT/DSEP/672		
Credits	4 Credits(2TH+2PR)	Contact Hours(TH)	2 Hrs / Week
		Contact Hours(PR)	4 Hrs / Week

Prerequisites:

- Basic understanding of programming concepts
- Familiarity with Python or another programming language
- Knowledge of fundamental machine learning concepts

Course Objectives(CO):

- Understand the principles and technologies behind AI chatbot services.
- Gain hands-on experience in designing and developing chatbot systems.
- Explore the applications and potential use cases of chatbot technology.
- Acquire knowledge of natural language processing (NLP) techniques for chatbots.
- Develop skills in training and deploying chatbot models using machine learning.

Learning Outcome(LO):

By the end of this course, students will be able to:

- Design and implement functional chatbot systems using AI techniques.
- Apply NLP methods to process and interpret user input in chatbot interactions.
- Utilize machine learning algorithms to train and improve chatbot models.
- Evaluate and choose appropriate chatbot platforms and tools for specific applications.
- Consider ethical, legal, and privacy implications in chatbot development

Course Outline:

Unit1: Introduction to AI Chatbot Services- Overview of AI chatbots and their significance, Historical development of chatbots, Applications and benefits of AI chatbot services, Introduction to natural language processing (NLP) and machine learning (ML) in chatbot development. Chatbot Design and Architecture- Chatbot design principles and user experience considerations, Conversational flow and dialogue management, Backend architecture and integration with existing systems, Introduction to chatbot development platforms and tools

Unit2: Natural Language Processing (NLP) Fundamentals-

Introduction to NLP and its role in chatbot services, Text preprocessing and tokenization, Named entity recognition (NER) and sentiment analysis, Word embeddings and language models, Machine Learning for Chatbot Development- Introduction to machine learning algorithms in chatbot development, Supervised, unsupervised, and reinforcement learning techniques, Training data collection and annotation, Evaluation metrics for chatbot performance

Unit3: Building Rule-Based Chatbots- Rule-based chatbot development approach, Designing rule-based chatbot architectures, Implementing intent recognition and entity extraction using rule-based methods, Limitations and challenges of rule-based chatbots. Introduction to Conversational AI - Understanding conversational AI and its components, Introduction to intent recognition and dialogue management systems, Dialog flow and other conversational AI platforms, Creating intents, entities, and dialogues in conversational AI platforms. ek 10: Advanced Chatbot Features and Techniques- Multilingual chatbot development, Emotion detection and sentiment analysis in chatbots, Contextual understanding and maintaining conversation context, Implementing voice-based chatbot interfaces

Recommended Reference Books:

"Chatbots: An Introduction and Easy Guide to Building Your Own" by Matthew Harper

"Designing Bots: Creating Conversational Experiences" by Amir Shevat

"Building Chatbots with Python: Using Natural Language Processing and Machine Learning" by Sumit Raj

"Conversational AI and Chatbots: Fundamentals, Applications and New Trends" by Manoj Prasad and Amitava Dasgupta

"Applied Artificial Intelligence: A Handbook for Business Leaders" by Mariya Yao, Adelyn Zhou, and Marlene Jia