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 Department)

(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 in Information Technology

A) Prologue

Welcome to Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad. The department is one of the most vibrant departments on the university campus and also recognized by Department of Science and Technology (**DST**) **FIST**, University Grants Commission (**UGC**) **SAP** (DRS – PHASE 1, PHASE 2) programs of Government of India.

Department of Computer Science and Information Technology adopted a credit- based system under the Academic Flexibility Program of the University from the academic year 2011- 12 and marching ahead with incorporation of guidelines of National Education Policy 2020. The department of computer science and information technology has provided excellent learning environment for the student by providing state of art curriculum offering latest technology trends that designed to meet expectation of industries and research. The courses offer flexible, cafeteria- type learning system with an inbuilt horizontal mobility for students to all desire units of education in the Department/Departments with provision for even inter Departmental mobility for students.

The Outcome Based Education (OBE) and Choice Based Credit System (CBCS) operating cohesively towards implementation of modular pattern where module / units called "credits", wherein 'credit' defines the quantum of contents / syllabus prepared for a course / paper and determines the minimum number of teachings- learning hours required to utilized towards module.

OBE & CBCS permits students to:

- Learn at their own pace,
- Choose electives from a wide range of elective courses offered by the department,
- Undergo additional/value added courses and acquire more than the required number of credits, depending
- upon the learner aptitude,
- Adopt an interdisciplinary approach in learning,
- Make best use of the expertise of faculty across the Department, beside the particular department faculty
- Acquire knowledge, skill and attitude of learning outcomes through participatory teaching and learning and continuous evaluation process

This provides the flexibility to make the system more responsive to the changing needs of our students, the professionals and society. The credit- based system also facilitates the transfer of credits.

B) Master's programs offered by the Department

Sr. No	Name of Master Program	Duration	Intake
1	M.Sc. Computer Science	1 Yr.[@] / 2Yr.[$]	32[*]+ 08[#]
2	M.Sc. Information Technology	1 Yr.[@] / 2Yr.[$]	32[#]
3	M.Sc. Artificial Intelligence	1 Yr.[@] / 2Yr.[$]	32[#]

*Grant in Aid, [#]Non-Grant, [@]PG-Diploma (up on exit), [$]Master Degree upon Exit option of NEP

C) Admission to M.Sc Information Technology Program

The admission to M.Sc. Information Technology program is conducted by the university by announcement of admission notification on www.bamu.ac.in and Centralized Admission process have been adopted for filling all seats as per intake capacity. For more information and detail kindly visit university website for all required details. Once the student is admitted to the department for the course, he/she will be promoted to next semester with full carryon; subject to the registration of student in every consecutive semester. Dropout student will be allowed to register for respective semester as and when the concerned courses are offered by the department, subject to the condition that his/her tenure should not exceed more than twice the duration of course from the date of first registration at parent department. The admission of concern student will be automatically get cancelled if he/she fails to complete the course in maximum period (Four years / Eight semesters) and to be observed with time to time amendments.

D) 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)

E) Course Fees:

Please refer to the course prospectus of university for the course fees. Course Fees per semester for M.Sc. Information Technology is 26200/- (Non-Grant) per year.

F) 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 department 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 Department has framed the curriculum as per the Model course structure suggested by NEP 2020 guidelines.

G) 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.

H) Attendance

M.Sc. Information Technology is full time course; therefore, it is necessary for the student to attend all theory as well as practical schedule precisely. The students must have 75% of attendance in each discipline specific core courses (DSC), discipline specific elective courses (DSE), skill courses and research methodology courses for appearing the internal evaluations (IE) and/or external evaluation examination. However, student having 65% attendance with medical certificate may apply to the Head of Department for commendation of attendance. The student failing to produce such permission or failing in achieving the required attendance, he/she will not be allowed to submit examination forms or attend the examination.

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 (MajorMandatory)-DSC	UDIT/MJT/500	Data Structure	2	-	2	-	14
	UDIT/MJT/501	Web Technologies	2	-	2	-	
	UDIT/MJT/502	Mathematical Foundations	2	-	2	-	
	UDIT/MJP/500	Practical based on Data Structure	-	4	-	2	
	UDIT/MJP/501	Practical based on Web Technologies	-	4	-	2	
	UDIT/MJP/502	Practical baesd on Mathematical Foundations	-	4	-	2	
Skill/Advance Course	UDIT/MJP/503-507	Programming-1*	-	4	-	2	2
Discipline Specific Electives - DSE	UDIT/DSET/520-523	Elective-1#	2	-	2	-	4
	UDIT/DSEP /520-523	Practical baesd on Elective-1#	-	4	-	2	
Research Methodology	UDIT/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 (MajorMandatory)-DSC	UDIT/MJT/550	Advanced Software Engineering	2	-	2	-	14
	UDIT/MJT/551	Machine Learning	2	-	2	-	
	UDIT/MJT/552	Advanced RDBMS	2	-	2	-	
	UDIT/MJP/550	Practical based on Advanced Software Engineering	-	4	-	2	
	UDIT/MJP/551	Practical based on Machine Learning	-	4	-	2	
	UDIT/MJP/552	Practical baesd on Advanced RDBMS	-	4	-	2	
Skill/Advance Course	UDIT/MJP/553-557	Programming-2*	-	4	-	2	2
Discipline SpecificElectives - DSE	UDIT/DSET/570-573	Elective-2#	2	-	2	-	4
	UDIP/DSEP/570-573	Practical baesd on Elective-2#	-	4	-	2	
On-Job Training	UDIT/OJT/590	On-Job Training / Field Project	-	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 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 (MajorMandatory)-DSC	UDIT/MJT /600	DataVisualization	2	-	2	-	14
	UDIT/MJT /601	Data Mining and Data Warehousing	2	-	2	-	
	UDIT/MJT /602	Internet of Things (IoT)	2	-	2	-	
	UDIT/MJP /600	Practical based on DataVisualization	-	4	-	2	
	UDIT/MJP /601	Practical based on Data Mining and Data Warehousing	-	4	-	2	
	UDIT/MJP /602	Practical baesd on Internet of Things (IoT)	-	4	-	2	
Skill/Advance Course	UDIT/MJP /603-607	Programming-3*	-	4	-	2	2
Discipline SpecificElectives - DSE	UDIT/DSET /620-623	Elective-3#	2	-	2	-	4
	UDIT/DSEP /620-623	Practical baesd on Elective-3#	-	4	-	2	
RP	UDIT/RP/649	Research Project-1	-	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 (MajorMandatory)-DSC	UDIT/MJT /650	Introduction to Quantum Computing	2	-	2	-	14
	UDIT/MJT /651	Artificial Intelligence	2	-	2	-	
	UDIT/MJT /652	Big Data Analytics	2	-	2	-	
	UDIT/MJP /650	Practical based on Introduction to Quantum Computing	-	4	-	2	
	UDIT/MJP /651	Practical based on Artificial Intelligence	-	4	-	2	
	UDIT/MJP /652	Practical baesd on Big Data Analytics	-	4	-	2	
Discipline SpecificElectives - DSE	UDIT/DSET /670 – 673	Elective-4#	2	-	2	-	4
	UDIT/DSEP /674-677	Practical baesd on Elective-3#	-	4	-	2	
RP	UDIT/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 UDIT/MJP/503-507	Programming-2 UDIT/MJP/553-557	Programming -3 UDIT/MJP/603-607
Java Group	Core Java	Advance Java	Android
Microsoft Group	Advanced C++	VB.NET	C# NET
Open Group	Python	Advanced Python	Open Web Programming (PHP)
Developer Group	Rust	Go	Kotlin
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** UDIT/DSET/520-523 UDIT/DSEP/520-523	**Elective 2** UDIT/DSET/570-573 UDIT/DSEP/570-573	**Elective 3** UDIT/DSET/620-623 UDIT/DSEP/620-623	**Elective 4** UDIT/DSET/670-673 UDIT/DSEP/670-673
Pattern Analysis & Machine Intelligence	Soft Computing	Fuzzy Systems : Theory, Application & Case Study	Video Processing	Pattern Recognition
Remote Sensing and Geospatial Technology	Fundamental of Satellite Remote Sensing	GIS	Remote Sensing And Digital Image Analysis	Hyperspectral Image Analysis
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) CourseCode(PR)	UDIT/MJT/ 500 UDIT/MJP/500	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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)	UDIT/MJT/ 501	Course Type	**Mandatory**
CourseCode(PR)	UDIT/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/TLSHTT Prequestmessage- responsemessage-WebClientsWebServers-CaseStudy,Cookies,Server-Side:DynamicContent,Web Content Management Systems (WebCMS).

Reference Books**:**

1. WebProgramming,buildingintcrnctapplications,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'.w3sclaools.coin/html/default.asp
3. https:.'/hiedrttartnalytics.nit.edu/
4. http://news.n1it.cdli/topic/web-deveIognJent

CourseTitle	**Mathematical Foundations**		
CourseCode(TH) CourseCode(PR)	UDIT/MJT/502 UDIT/MJP/502	CourseType	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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,Venndiagrams,statementproblems,AssociativeLaws,Distributivelaws.DeMorganslaws, 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 arelation, 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. KennethHRoaen(IndianAdaptationByKamalaKrithivasan),DiscreteMathematicsAndItsApplications WithCombinatoricsAndGraphTheory,seventhedition,McGrawHillEducation.(Unit1)
2. CLLiu,DPMohapatra,"ElementsofDiscreteMathematics"3rdedition,McGrawHill,2008..(Unit1)
3. KothariC.R.&GargGaurav,(2019),ResearchMethodologyMethods&Techniques(fourthEdition), New Age International Publishers, New Delhi. (Unit2-5)
4. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rdEdition, Tata McGraw Hill.
5. KolmanandBusby—DiscreteMakematicalstructuresforComputerSciencesPm.

CourseTitle	**Research Methodology (Information Technology)**		
CourseCode	UDIT/RM/530 Course Type		**Mandatory**
Credits	**4 Credits**	Contact Hours(TH)	**4 Hrs/ Week**

Prerequisites:

- Basic understandingof computerscienceconcepts
- Familiaritywith programming languages(e.g.,Python,R,orMATLAB)
- Knowledge of introductory statistics

Course Objectives(CO):

- To introduce studentst of the fundamental principles of research methodology in computer science.
- To provide students with a comprehensive understanding of computational and statistical methods usedin computerscience 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 techniquesfor hypothesis testing and inference.
- To familiarize students with machine learning algorithm and their applications in computer science research.
- To develop students'ability todesign and analyse experiments in the context of computersciencer research.
- To enhance students'critical thinking and problem-solving skills through cases studies and practical assignments.

Learning Outcomes (LO):

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

- Formulateresearchquestionsandselectappropriateresearchmethodologies.
- 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 incomputer scienceresearch.
- Critically evaluate research studies and identify strengths and limitations int their methodology.

CourseOutline:

Unit 1: Introduction to Research Methodology- Overview of researchmethodology 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 andsummarizingreal-worlddatasets

Unit 3: Statistical Analysis - Descriptive statistics: measures of central tendency, dispersion, etc., Hypothesistesting 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

Unit5: 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.

AssessmentMethods:

- 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. "ResearchMethodology:AStep-by-StepGuidefor Beginners"byRanjit Kumar
2. "DesigningandConductingMixedMethodsResearch"byJohnW.CreswellandVickiL.PlanoClark"StatisticalMethodsforComputer Science"byWalter D.Wallis
3. "Pattern Recognition and Machine Learning" by Christopher M. Bishop"DataScienceforBusiness"byFosterProvostandTomFawcett
4. "TheElementsofStatisticalLearning:DataMining,Inference,andPrediction"byTrevorHastie,RobertTibshirani,andJerome Friedman
5. "VisualizationAnalysisandDesign"byTamaraMunzner

Semester–II

CourseTitle	**Advanced Software Engineering**		
CourseCode(TH) CourseCode(PR)	UDIT/MJT/ 550 UDIT/MJP/550	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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) CourseCode(PR)	UDIT/MJT/ 551 UDIT/MJP/ 551	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basic knowledge of probability and statistics
- Familiarity with programming (Pythonpreferred)
- 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 andunsupervised learning, Model evaluation and selection. Regression Analysis - Linear regression, Polynomialregression, 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.,RandomForest,GradientBoosting), Casestudy:Spamemailclassification.

Unsupervised Learning- Clustering algorithms (e.g., K-means, Hierarchicalclustering), Dimensionality reduction techniques(e.g.,PCA,t-SNE) Case study:Customersegmentation

Unit 3: Deep Learning Fundamentals - Neural networks and deep learning architectures, Convolutional NeuralNetworks(CNNs)forimageanalysis,RecurrentNeuralNetworks(RNNs)forsequentialdata,Casestudy:Imageclassification usingCNNs.

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 theCIFAR-10 dataset usingCNNs.
3. Movie Recommender System: Developing are commendation engineusing 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 fore casting 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" byIan Goodfellow, Yoshua Bengio, and Aaron Courville,

CourseTitle	**Advanced RDBMS**		
CourseCode(TH) CourseCode(PR)	UDIT/MJT / 552 UDIT/MJP /552	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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) CourseCode(PR)	UDIT/MJT / 600 UDIT/MJT /600	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basic understanding of statistics and data analysis concepts
- Familiarity with spreadsheet software (e.g.,MicrosoftExcel,GoogleSheets)
- 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 in corporating 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 acomprehensive 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 in formative data visualizations.
- Develop in teractive visualizations to facilitate data exploration and use rengagement.
- Apply data visualizations kills to analyze and present complex data sets effectively.
- In corporate story telling elements into data visualizations to convey in sights 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 annotations 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) CourseCode(PR)	UDIT/MJT /601 UDIT/MJP /601	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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
- Ga in 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 in sights 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 indecision-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 datawarehouse, including ETL processes and dimensional modeling.
- Utilize OLA Ptools for data analysis and reporting.
- Createcompellingdatavisualizationsandinteractivedashboards.
- Analyse real-world case studies to solve business problems using data mining and warehousingtechniques.

CourseOutline:

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,Matplot lib). 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), Evaluationmetrics 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, snow flake 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 tail or 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.

"DataWarehousing Fundamentals"by Paulraj Ponniah.

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

CourseTitle	**Internet of Things(IoT)**		
CourseCode(TH) CourseCode(PR)	UDIT/MJT/ 602 UDIT/MJP /602	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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 datamanagement 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, andanalyzing IoTdata
- To examinereal-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 solution susing appropriate platforms and frame works

Learning Outcomes (LO):

By the end of this course, students will be ableto:

- 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 relevantplatformsand 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

CourseOutline:

Unit 1: Introduction to IoT - Definition, history, and evolution of IoT, Key components of an IoT system, IoT ecosystem and stake holders. 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 computingand 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 andCities-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 YourIndustry" 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) CourseCode(PR)	UDIT/MJT/ 650 UDIT/MJP/ 650	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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 (super conducting qubits,trappedions,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 TetsuoOhmi "
Quantum Computing: An Applied Approach" byJackD. Hidary

CourseTitle	**Artificial Intelligence**		
CourseCode(TH) CourseCode(PR)	UDIT/MJT / 651 UDIT/MJP /651	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basic programming knowledge (Pythonpreferred)
- Understanding of linear algebraand calculus
- Familiaritywithstatisticsandprobabilitytheory

Course Objectives (CO):

- Gain a practical understanding of the core concepts and technique in Artificial Intelligence.
- Develop theskills 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 A Itechniques to real-worldproblemsthroughcasestudies.
- 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 variousAI tasks.
- Apply NLP technique to analyse and process textual data.
- Utilize computer visional agorithms for image analysis and object trecognition.
- Understand the principles of reinforcementlearning and develop RL-based agents.
- Analyse and interpret AI case studies in different domains.
- Demonstrate an awareness of ethical considerationsi nAI and theimpact of AI n society.

Course Outline:

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

Unit 2: Neural Networks and Deep Learning - Introduction to neural networks, Convolution Neural Networks(CNNs),RecurrentNeuralNetworks(RNNs),Deeplearningframeworks(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, QuantumAI) , 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) AIinEducation-Personalized educationwith AI,g) AI in Governance

#use of additional learning resource is highly appreciated.

Recommended Reference Books:

1. "Artificial Intelligence:AModernApproach"by StuartRusselland Peter Norvig"Deep Learning"byIan Goodfellow,YoshuaBengio,andAaronCourville
2. "NaturalLanguageProcessingwithPython"byStevenBird,EwanKlein,andEdwardLoper"ComputerVision:AlgorithmsandApplications"byRichardSzeliski
3. "Reinforcement Learning:AnIntroduction"byRichardS.SuttonandAndrewG.Barto
4. "The Hundred-Page MachineLearning Book"byAndriy Burkov

CourseTitle	**Big Data Analytics**		
CourseCode(TH) CourseCode(PR)	UDIT/MJT / 652 UDIT/MJP /652	Course Type	**Mandatory**
Credits	**4Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **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, andtools.
- To develop practical skills for data collection, pre-processing, and exploratory analysis.
- To introduce students to various datamining 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 analys is techniques to gain in sights and visualize data effectively.
- Utilize different datamining techniques for pattern discovery, clustering, and classification.
- Understand the challenges and opportunities associated with big dataanalytics.

CourseOutline

Unit 1: Introduction to Data Analytics - Understanding the importance of data analytics, Overview of dataanalytics lifecycle, Introduction to data analysis tools and technologies. Data Preparation and Cleaning - Datacollection and pre-processing techniques, Handling missing data and outliers, Data transformation and featureengineering. 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 andclassification algorithms, Model evaluation and selection. Time Series Analysis - Understanding time series data,Trend analysis and seasonality, Forecastingtechniques.ClusteringandDimensionalityReduction-Unsupervisedlearningalgorithms,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 andfeature 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 AnalyticsforBeginners"byJohnSmith
2. "DataScienceforBusiness"byFosterProvostandTomFawcett"Python forDataAnalysis"byWesMcKinney
3. "StatisticsforDataScience"byJamesMiller
4. "PredictiveAnalytics:ThePowertoPredictWhoWillClick,Buy,Lie,orDie"byEricSiegel "Forecasting: Principles and Practice" by Rob J. Hyndman and George Athanasopoulos"Pattern Recognition andMachineLearning" by ChristopherM.Bishop

****

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

ProgrammingGroup	Programming-1 UDIT/MJP/503-507	Programming-2 UDIT/MJP /553-557	Programming-3 UDIT/MJP /603-607
Java Group	Core Java	Advance Java	Android
MicrosoftGroup	AdvancedC++	VB.NET	C#NET
Open Group	Python	Advanced Python	Open Web Programming (PHP)
DeveloperGroup	Rust	Go	Kotlin
Web Scripting 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 thecourse.

Programming Group:Java Group

Course Title	**Core Java**		
CourseCode(PR)	UDIT/MJP /503	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**		

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-OrientedProgramming (OOP) Basics - Classes and objects, Encapsulation, Inheritance, and Polymorphism, Methodoverloading andoverriding, Constructorsandstaticmembers,Accessmodifiers

Unit 2: Advanced OOP Concepts - Abstract classes and interfaces, Packages and access control, Exceptionhandling, 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 bytestreams, Reading and writing files, Serialization and deserialization, Working with directories, Multithreading -Understanding threads, Creating and running threads, Synchronization and thread safety, Thread communicationand coordination,Threadpoolsandexecutors

Unit 3: GUI Programming with Applet and Swing – Life Cycle of Applets & Swings components, Creating UIcomponents – Applet & Swings (buttons, labels, text fields, etc.), Event-driven programming, Layout managers,CreatingDialogs and message boxes, Swing Layouts.Revisitingto Databases and SQL - Understanding databases and database management systems, Introductionto 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 JDBCConcepts-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 laboratorywork.

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 studentduringthe course work with Code optimization and best practices.

Recommended Reference Books:

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

Course Title	**Advance Java**		
CourseCode(PR)	UDIT/MJP /553	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**		

Prerequisite:

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

Course Objectives (CO):

- Develop a deep understanding of advanced Java webdevelopment concepts
- Gain proficiency in using Servlets,Struts,and Hibernate frameworks
- Learn to design and build scalable and maintainable web applications
- Acquire hands-onexperiencei n 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 frame works 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

CourseOutline:

Unit 1: Introduction to Advanced Java Web Development - Overview of Java Servlets, Struts, and Hibernate,Comparisonofdifferentwebframeworks,UnderstandingtheMVC(Model-View-Controller)architecture.Servlets: Building Dynamic Web Applications - Servlet life cycle and request handling, Handling form data andrequest parameters, Session management and authentication, Servlet filters and listeners, Error handling andexception management

Unit 2: Struts Framework: Structured Web Application Development - Introduction to Struts framework and itscomponents, Configuring Struts and defining action mappings, working with forms and validation, managingdatabase operations using Struts, Implementing security and authentication in Struts. Hibernate: Object-RelationalMapping (ORM) - Introduction to Hibernate and ORM concepts, Configuring Hibernate with different databasesystems, Mapping Java objects to database tables, Performing CRUD operations using Hibernate, Querying datausing Hibernate QueryLanguage (HQL)

Unit 3: Integration of Servlets, Struts, and Hibernate - Leveraging the power of Servlets, Struts, and Hibernatetogether, building a complete end-to-end web application, Implementing layered architecture and separation ofconcerns, Handling transactions and database operations. Case Studies a) Building ane-commerce platform using Servlets, Struts, and Hibernate, b)Develop in gasocial 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 frame works

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. "Hibernatein Action"by Christian Bauerand Gavin King
4. "Pro Spring MVC: With Web Flow"by Marten Deinum,Koen Serneels, and ColinYates
5. "Java Persistence with Hibernate"by Christian Bauerand Gavin King

Course Title	**Android**		
CourseCode(PR)	UDIT/MJP /603	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**		

Prerequisite:

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

Course Objectives (CO):

- Understand the fundamental concepts and architecture oft the Android platform.
- Gain proficiency in developing Android applications using Java.
- AcquireknowledgeofvariousAndroidcomponentsandtheirfunctionalities.
- Learn todesignuser-friendlyandvisuallyappealingAndroidinterfaces.
- Develop skills in handling data storage, network communication, and multimedia integration inAndroid applications.
- Apply bestpracticesandcoding standardsfordevelopinghigh-qualityAndroidapps.
- Gainhands-onexperiencebyworkingonreal-worldcasestudies.
-

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 technique ssuch 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: Microsoft Group

Course Title	**Advance C++ Programming**		
CourseCode(PR)	UDIT/MJP /504	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 C++ programming language fundamentals
- Knowledge of basic data structures and algorithms
- Familiarity with object-oriented programming concepts

Course Objectives (CO):

- To deepen students' understanding of C++ programming language features and syntax.
- To provide students with advanced techniques for solving complex programming problems.
- To enhance students' skills in designingfficient and maintainable C++ code.
- To introduce students to key concepts in object-oriented design and software architecture.

Learning Outcomes (LO):

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

- Apply templates and generic programming to creature usable and type-independent code.
- Implement effective exception handling strategies to enhance program robustness.
- Manage memory dynamically and prevent common memory-relatedissues.
- Utilize advanced data structures to solve complex problems efficiently.

- Design and implement object-oriented solutions using inheritance, polymorphism, and designpatterns.
- Analyze and refactor existing code to adhereto SOLID principles and improvecode quality.

CourseOutline:

Unit 1: Advanced Language Features - Templates and Generic Programming, Lambda Expressions and Closure, Move Semantics and Perfect Forwarding, Smart Pointers and Resource Management, Variadic Templates. Object-Oriented Programming Techniques - Inheritance and Polymorphism, Virtual Functions and Abstract Classes, Multiple Inheritance and Interface Design, Run-TimeType Information (RTTI), Object Slicing and Virtual

Destructors.TemplatesandGenericProgramming-Functiontemplatesandclasstemplates, Template specialization and partial specialization,Advanced template techniques

Unit 2: Exception Handling and Error Management - Exception Handling Basics, Custom Exception Classes,Resource Acquisition Is Initialization (RAII) and Exceptions, Error Handling Strategies and Best Practices.MemoryManagement-Dynamicmemoryallocation,Resourcemanagementusingsmartpointers,Memoryleaksand memorycorruption,Garbage collection

Unit 3: Advanced Data Structures and Algorithms- Standard Template Library (STL) Overview, CustomContainer Classes, Advanced Algorithms and Data Manipulation, Sorting and Searching Techniques, GraphAlgorithms and Traversal. Object-Oriented Design Principles - Inheritance and polymorphism, Abstract classesand interfaces, Design patterns, SOLID principles Case Studies and Real-World Applications a) Developing aTextProcessingApplication, b)Implementinga Database Management System, c)CreatingaNetworkingLibrary

a) Building a Game Engine,e)Developing a Compiler or Interpreter

RecommendedReferenceBooks:

1. "Effective Modern C++"by ScottMeyers
2. "C++Primer"by Stanley B.Lippman,JoséeLajoie,and Barbara E.Moo
3. "TheC++Programming Language"byBjarneStroustrup
4. "EffectiveC++"byScott Meyers
5. "ModernC++Design: Generic Programming and Design PatternsApplied"byAndreiAlexandrescu
6. "C++Concurrency in Action: Practical Multithreading"by Anthony Williams
7. "AdvancedC++Meta programming"by Davide DiGennaro

Course Title	**VB. NET Programming**		
CourseCode(PR)	UDIT/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**		

Prerequisite:

Basic programming knowledge and familiarity with concepts like variables, data types, and control structureswould bebeneficial. NopriorexperiencewithVB.NETisrequired.

Course Objectives (CO):

- To develop a strong foundation in VB.NET programming concepts and syntax.
- To understand the principles of object –oriented programmingandtheirimplementationinVB.NET.
- TogainproficiencyinGUIdevelopmentusingWindowsFormsandevent-drivenprogramming.
- TolearnfileI/OoperationsanddatabaseprogramminginVB.NET.
- Toexploreadvancedtopicssuchasmultithreading, error handling, andLINQ.

Learning Outcomes (LO):

Bytheend ofthiscourse, students will be ableto:

- Writewell-structuredVB.NETprogramsusingappropriatesyntaxandprogrammingconstructs.
- DesignanddevelopinteractiveWindowsFormsapplicationswithagraphicaluserinterface.
- ImplementfileI/Ooperationsforreadingfromand writingtofiles.
- Perform database operationsusingADO.NETand SQL.
- Applyerrorhandlinganddebuggingtechniquesto identifyand fixissuesinVB.NETcode.
- Demonstrate an understanding of advanced topics like multithreading, XML/JSON processing, andLINQ.

CourseOutline:

Unit 1: Introduction to VB.NET - Overview of VB.NET and its evolution, Understanding the .NET Framework,Setting up the development environment, VB.NET syntax and basic program structure. Data Types, Variables, andOperators - Built-in data types in VB.NET, Declaring and using variables, Arithmetic, comparison, and logicaloperators, Working with strings and string manipulation. Control Structures and Decision Making - Conditionalstatements (if-else, switch), Looping structures (for, while, do-while), Jump statements (break, continue, return),Error handling and exception handling

Unit2:Object-OrientedProgramminginVB.NET-Understandingobject-orientedprogramming(OOP)concepts,Classes,objects,andinstances,Inheritanceandpolymorphism,Encapsulationanddatahiding,Constructors, destructors, and properties. GUI Development with Windows Forms - Introduction to WindowsForms applications, Designing user interfaces with controls and layouts, Handling events and event-driven programming,Working withmenus,dialogs,and commoncontrolsDatabinding and validation

Unit 3: File Handling and Database Connectivity - Reading from and writing to files, Working with directoriesand file operations, Introduction to databases and ADO.NET, Connecting to databases and executing queries,Performing CRUD operations on databases, Advanced Topics in VB.NET - Multithreading and asynchronousprogramming, Working with XML and JSON data, Introduction to LINQ (Language Integrated Query), Web services and APIintegration, Deployment and application publishing, Bestpractices and performance optimization. Case Studies: Throughout the course,several case studies will be presented to reinforce the learning objectives and provide practical applications of VB.NET programming. These case studies may include a) Building a library management system, b) Developing a payroll management system, c) Creating a customer relationship management(CRM) application,d) Designing as implegame

using Windows Forms

RecommendedReferenceBooks:

1. "VisualBasic.NETProgrammingforBeginners"byJohnSmiley
2. "MasteringVisualBasic.NET"byEvangelosPetroutsos
3. "ProgrammingVisualBasic.NET"byJesseLiberty
4. "VB.NETCoreClassesinaNutshell"byBudiKurniawan
5. "VisualBasic.NETLanguagePocketReference"byStevenRoman

Course Title	**C # Programming**		
Course Code(PR)	UDIT/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:

- Basicunderstandingofprogrammingconcepts
- FamiliaritywithaprogramminglanguageCorJava)
- Knowledgeoffundamentalcomputersciencepprinciples

Course Objectives(CO):

- Develop a strong foundation in C#programming language
- Understandobject-orientedprogramming(OOP)conceptsandapplythemeffectively
- GainpracticalexperienceindevelopingsoftwaresolutionsusingC#
- Acquireproblem-solvingskillsthroughcasestudiesandprogrammingchallenges
- PrepareforadvancedC#programmingorrelatedcourses

Learning Outcomes(LO):

Bytheend ofthiscourse,students will be ableto:

- WriteandexecuteC#programstosolvereal-worldproblems
- Applyobject-orientedprogrammingprinciplesinC#development
- Designanddevelopgraphicaluserinterfaces(GUIs)usingWindowsFormsorWPF
- ImplementexceptionhandlingmechanismsinC#applications
- Demonstrateproficiencyinworkingwitharrays,collections,andfileI/Ooperations
- AnalyzeandsolveprogrammingchallengesusingC#programmingtechniques

Course Outline:

Unit1:IntroductiontoC#Programming-IntroductiontoC#anditsroleinsoftwaredevelopment,Settingupthedevelopment environment (IDE, compiler, etc.), Basic program structure and syntax, Variables, data types, andtype conversions, Input and output operations. Object-Oriented Programming with C# - Understanding object-oriented programming (OOP)concepts, Classes ,objects, and methods, Encapsulation,inheritance,andpolymorphism,Constructors and destructors, Access modifiers and properties. Control Structures and Flow of Execution-Conditional statements (if-else, switch-case), Looping structures(for,while,do-while),Jump statements(break, continue,return), Exception handling and error management

Unit 2: C# Data Structures and Collections - Arrays and lists, Dictionaries and hash tables, Stacks and queues,Enumerations and tuples.File Handling and Input/OutputOperations-Reading and writing files,Working with directories and file paths,Stream-basedinput/output operations

Unit 3: Advanced C# Programming Concepts - Delegates and events, Generics and generic collections, LINQ(Language-IntegratedQuery),Multithreading and asynchronous programming. C# Application DevelopmentCaseStudies:a)Building as impleconsole application, b)Developing a Windows Formsapplication, c)Creating a WPF (Windows Presentation Foundation) application, d) Developing a web application with ASP.NET

Recommended Reference Books:

1. "C#9and.NET5–ModernCross-PlatformDevelopment"byMarkJ.Price

2. "C#9.0inaNutshell:TheDefinitiveReference"byJosephAlbahariandBenAlbahari

3. "ProC#9with .NET5"byAndrewTroelsenand Philip Japikse

"Head First C#: A Learner's Guide to Real-World Programming with C#, XAML, and .NET" by JenniferGreeneandAndrew Stellman

4. "C#ProgrammingYellowBook"byRobMiles

Programming Group: Open Group

Course Title	**Python Programming**		
CourseCode(PR)	UDIT/MJP /505	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:

- Thereare no specific prerequisites for this course. However, abasicunder standing 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.
- Analyse and debug Python code for errors and exceptions.
- Develop a basic understanding of GUI programming using Tkinter.

Course Outline:

Unit1: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), Controlflow andbranching

Unit2:FunctionsandModules-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 toGUI 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 graphicaluserinterfaces(GUI) fordesktop applications.

Recommended Reference Book:

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

Course Title	**Advance Python Programming**		
CourseCode(PR)	UDIT/MJP /555	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):

- Developadeep 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 mtooptimize performance.
- Enhance debugging and testings killstoen sureth 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 Python iccode following best practices.
- Effectively debug and test Python applications for identifying and fixing issues.
- Apply the learned concept storeal-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, Designpatterns and their implementation in Python, Advanced OOP concepts: abstract classes, interfaces, and multipleinheritance, Functional Programming in Python - Higher-order functions, lambda expressions, and closures,Immutable data structures and pure functions, Functional programming techniques: map, filter, reduce, andrecursion

Unit3: Concurrency and Parallelismin Python-Multithreading and multiprocessing, Synchronization and thread safety, Asynchronous programming with async/await. Python Database Libraries - Overview of popular Python data base libraries (e.g.,SQL Alchemy, psycopg2), Installation and setup of data base libraries, connecting to data bases using Python. Database Querying with Python - Executing SQL queries using Python, Fetching and manipulating query results,Parameterized queriesand prepared statements, Object-Relational Mapping(ORM)-Introduction to ORM frameworks (e.g., SQLAlchemy), 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 participant stotackle 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"byEricMatthes

"Pythonfor DataAnalysis" byWes

Course Title	**Open Web Programming (PHP)**		
CourseCode(PR)	UDIT/MJP /605	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**		

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 ProgramminginPHP-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-sidescripting and client-server communication, Handling form (GET, POST) submissions, Session management andcookies. 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: Developer Group

Course Title	**Rust**		
CourseCode(PR)	UDIT/MJP /506	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**		

Prerequisite:

Priorprogramming experience in any language (e.g., Python, Java,C++) is recommended. Familiarity with basic programming concepts such as variables, control flow, and functions is beneficial. Noprior knowledge of Rustis required.

Course Objectives(CO):

- Introduce participants to the Rust programming language and its unique features.
- Provide a solid understanding of Rust syntax, data types, and control flow.
- Familiarize participants with error handling techniques in Rust.
- Explore advanced concepts like concurrency, parallelism, and asynchronous programming in Rust.
- Enable participants to build real-world applications using Rust.
- Develop problem-solving skills through case studies and hands-on programming exercises.

Learning Outcomes(LO):

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

- Write Rust programs using the correct syntax and idiomatic style.
- Apply error handling techniques to handle and propagate errors effectively.
- Develop concurrent and parallel applications using Rust'sconcurrency primitives. Implement advanced features such as generics, traits, and macros in Rust programs.
- Analyzeand solve programming problems using Rust as the primary language.
- Createreal-world applications by leveraging Rust's performance and safety guarantees.

Course Outline:

Unit 1: Introduction to Rust - Introduction to Rust and its key features, Installation and setup of the Rust development environment, Basic syntax, data types, and variables in Rust, Control flow and loops in Rust, Functions and modules in Rust. Ownership, Borrowing, and Lifetimes - Ownership principles and memory management in Rust, Borrowing and references in Rust, String and vector manipulation in Rust, Error handling and exception control in Rust, Handling errors with the Result and Option types, Panicand un winding, The panic! And un wrap ()macros, Error propagation using? operator, Lifetimes and their significance in Rust programming

Unit 2: Structs, Enums, and Pattern Matching - Defining and using structs in Rust, Enumerations and their applications in Rust, Pattern matching and its role in Rust programming, Traits and generics in Rust, Error handling and Option type in Rust

Unit 3: Concurrency and Parallelism - Introduction to concurrent programming in Rust, Working with threads and synchronization primitives, Message passing and shared-state concurrency in Rust, Asynchronous programming with async/await in Rust, Error handling in concurrent Rust programs. Advanced Concepts andLibraries - Advanced features and idiomatic Rust programming, File I/O and working with external libraries inRust, Testing and debugging techniques in Rust, Performance optimization and benchmarking in Rust,Case studies and real-world applications of Rusta)Buildinga Web APIwithRustand Rocket Frame work: Designingand implementing a RESTful API using Rust and Rocket framework, Handling request routing, middleware, andauthentication, Incorporating database interactions with Diesel ORM, b) Developing a Multithreaded Web Scraper with Rayon: Creating a web scraper to extract data from multiple sources concurrently, Leveraging Rayon forparallelizing the scraping process, Managing thread synchronization and data sharing in Rust c) Building a Command-Line Tool for File Encryption: Implementing a command-line tool using Rust for encrypting files, Utilizing cryptographic libraries in Rust, Handling command-line arguments and user input validation

RecommendedReferenceBooks:

"The Rust Programming Language "by Steve Klabnikand Carol Nichols

"Programming Rust: Fast, Safe Systems Development "by Jim Blandy and Jason Orend or ff"Rustin Action" by Tim Mc

Namara

"Rust Cook book"by Vesa Kaihla virta

"Hands-On Concurrency with Rust"by BrianL.Trout wine

Course Title	**Go**		
CourseCode(PR)	UDIT/MJP /556	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:

- Basic understanding of programming concepts
- Familiarity with at least one programming language

Course Objectives(CO):

- Gain asolid understanding of the Go programming language and its key features.
- Learn best practices for writing efficient and scalable Go code.
- Develop the ability to designand implement programs in Go for various applications.
- Acquire skills in concurrent programming and error handling using Go.
- Build web applications and networked systems using Go.

Learning Outcomes(LO):

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

- Write and debug programs in Go using appropriate syntax and language features.
- Apply Go'sconcurrency primitives and develop concurrent applications.
- Implement error handling strategies and write robust Go code.
- Build web applications and interact with data bases using Go.
- Analyze and solve real-world programming challenges using Go.

CourseOutline:

Unit1: Introduction to GO Programming- Overview of GO programming language, Setting up the GO development environment, Hello World program in GO, GO tool chain and package management. GO Language Basics - Variables and data types in GO, Constants and enumerations, Operators and expressions, Control structures: if-else, switch, loops

Unit 2: Functions and Packages - Defining and invoking functions, Function parameters and return values,Variable scope and lifetime, Introduction to packages and imports.Data Structures in GO - Arrays, slices, andmaps, Structs and pointers, Working with strings and byte slices, File I/Ooperations

Unit3: Error Handling andTesting –Error handling in GO: errors package, panic, and recover, Writing unittestsin GO, Benchmarking and profiling GO code. Concurrency and Goroutines-Introduction to concurrent programming, Go routines and channels in GO,Synchronization and mutual exclusion, Error handling inconcurrent code, Case Studies and Project Development - Real-world case studies showcasing GO's strengths,Developing apracticalproject using GO, Best practices and code organization in GO

Recommended Reference Books:

"The Go Programming Language" by Alan A. A. Donovan and Brian W. Kernighan"ConcurrencyinGo:Tools and Techniques

for Developers"by Katherine Cox-Buday"GoinAction"by William Kennedy, Brian Ketelsen,and ErikSt.Martin

"Introducing Go: Build Reliable, Scalable Programs"by Caleb Doxsey

"Mastering Go: Create Golang production applications using network libraries, concurrency, and advanced datastructures"by Mihalis Tsoukalos

Course Title	**Kotlin**		
CourseCode(PR)	UDIT/MJP /606	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**		

Prerequisite:

No prior programming experience is required for this course. However, familiarity with basic programmingconceptsandlogic willbe beneficial.

Course Objectives(CO):

- Introduce students to the Kotlin programming language and its key features.
- Enable students to write efficient and clean code using Kotlin.
- Familiarize students with Kotlin's object-oriented programming concepts and functional programming paradigms.
- Develop students'ability to build Android applications using Kotlin.
- Provide students with practical experience through case studies and hands-on exercises.

Learning Outcome(LO):

By the end of the course, students will:

- Have as olid understanding of the Kotlin programming language.
- Beable to write Kotlin code for various applications, including Android and web development.
- Possess the skills to apply Kotlin's advanced features effectively.
- Be capable of buildingreal-world applications using Kotlin.
- Have gained the confidence to continue learning and exploring Kotlinin dependently.

CourseOutline:

Unit 1: Introduction to Kotlin - Overview of Kotlin programming language, Kotlin's features and advantages,Setting up the development environment, Writing and executing a simple Kotlin program, Basic data types,variables,andoperators.ControlFlowandFunctions-Conditionalstatements(if-else,when),Loopingstructures(for,while,do-while), Functionsand parameters, Returningvaluesfromfunctions, Recursivefunctions

Unit2:Object-OrientedProgrammingwithKotlin-Introductiontoobject-orientedprogramming(OOP)concepts, Classes, objects, and properties, Inheritance and polymorphism, Interfaces and abstract classes, Dataclasses and sealed classes.Collections and Generics - Working with arrays, lists, sets, and maps, Iterating andmanipulating collections, Introduction to generics and type constraints, Creating and using generic functions andclasses. Kotlin Advanced Features - Null safety, Extension functions and properties, Collections and functionalprogramming Coroutines for asynchronous programming Kotlin and Android Development - Interoperability withJava, KotlinAndroidExtensions,Working withAndroid Studio andKotlin

Unit 3: Exception Handling and File I/O - Understanding exceptions and error handling, Handling exceptionsusingtry-catchblocks,Throwingandcatchingcustomexceptions,Readingfromandwritingtofilesusing Kotlin,Kotlin Standard Library - Exploring the Kotlin Standard Library, Commonly used functions and extensionfunctions, Working with strings, dates, and times, File manipulation and I/O operations. Kotlin frameworks forweb development(e.g.,Ktor,Spring Boot),BuildingRESTfulAPIswithKotlin.CaseStudiesa)ApplyingKotlinconceptstoreal-worldscenarios,b)Developingcasestudiesinvolvingapplicationdevelopment,datamanipulation,orproblem-solving,c)Analyzing and implementing solutionsusingKotlinprogramming

RecommendedReferenceBooks:

"Kotlin in Action" by Dmitry Jemerov and Svetlana Isakova"Programming Kotlin" by Stephen Samuel and Stefan

Bocutiu"KotlinforAndroid Developers" byAntonio Leiva

"KotlinProgramming:TheBigNerdRanchGuide"byJoshSkeenandDavidGreenhalgh

ProgrammingGroup:Web Scripting Group

Course Title	**VB & Java Script**		
CourseCode(PR)	UDIT/MJP /507	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**		

Prerequisite:

No prior programming experience is required for this course. However, a basic understanding of computerconceptsand familiarity with using computers and the internetisre commended.

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 usingVB.
- 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 desk to apapplications 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.

CourseOutline:

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, Exceptionhandling and errortrapping, File handling and datainput/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 JavaScriptframeworks(e.g., React,Angular)

Unit 3: Case Studies: a) Building a Desktop Application with Visual Basic: Students will develop a desktopapplication using Visual Basic, incorporating various concepts covered in the course. The case study will focuson user interface design, data management,and mplementing business logic b) Creating Dynamic Web PageswithJavaScript:Studentswillcreateinteractivewebpagesusing JavaScript.Theywilllearntomanipulatethe

DocumentObject Model (DOM), handleevents, and retrieve data from external sources, show casing the power of Java Script in web development.

Recommended Reference Books:

"Visual Basic 2019 in 24 Hours, Sams Teach Yourself" by James Foxall"Murach'sJava Script and jQuery"by Zak Ruval

cabaand Mary Delamater

"Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke"JavaScript:The

GoodParts"byDouglasCrockford

"ProfessionalVisualBasic2019and.NETCore3.0"byBillSheldon,BillyHollis,andRobWindsor

Course Title	**Node.js**		
CourseCode(PR)	UDIT/MJP /557	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**		

Prerequisite:

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

Course Objectives (CO):

- Gain athorough 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 data base 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. jsandrelated frameworks
- Implement as ynchronous programming techniques to handle concurrent operations efficiently
- Integrate data bases and implementuser authentication in Node.js applications
- Apply best practices for testing, debugging, and deployment of Node.js applications
- Analyse 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 programmingconceptsandcallbacks, Promisesand async/await forhandlingasynchronousoperations

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 applicationframework. WorkingwithDatabases- 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-DataMapping)

Unit 3: Asynchronous Programming in Node.js - Understanding the Event Loop and non-blocking I/O, Usingcallbacks,Promises,andasync/awaitforasynchronousprogramming,Errorhandlingandbestpractices,Middleware and Authentication - Implementing middleware for request processing, User authentication andauthorizationtechniques,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 inreal-time CaseStudies 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: AHands-On Guide to BuildingWeb 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**		
CourseCode(PR)	UDIT/MJP /607	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:

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

Course Objectives (CO):

- Gainathorough 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 sof 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 Reacthooks 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 even thandling in React applications
- Create single-page applications with React Router
- Apply styling technique stoenhance the visual appeal of React applications
- Implement form handling and validation in React
- Understand and apply advanced React concepts for optimal performanceanderrorhandling

CourseOutline:

Unit1:IntroductiontoReact-OverviewofReactanditskeyfeatures,UnderstandingtheReactcomponentmodelSettingupadevelopmentenvironmentforReact, IntroductiontoJSX(JavaScriptXML).ReactComponentsandState Management - Creating functional and class components, understanding component lifecycle methods,managingcomponentstate withhooksand context,Handlingeventsanddata binding inReact

Unit2:ReactRoutingandNavigation-IntroductiontoReactRouterforhandlingclient-siderouting,Implementingnavigationandnestedroutesin React,Managingrouteparametersandquerystrings,Implementingdynamic 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 stylesin React

Unit3:AdvancedReactConcepts-StatemanagementwithReduxorMobX,integratingthird-partylibrariesandAPIs with React, Server-side rendering with React (Next.js), Testing and debugging React applications. ReactFormsandValidation:Controlledanduncontrolledcomponents,Formhandlingandvalidationtechniques,Formlibrariesandformikintegration,AdvancedReactConcepts:Reacthooksforcustomlogic(useReducer,useContext,etc.), Performanceoptimization techniques,Errorhandling and debugging inReact

RecommendedReferenceBooks:

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

Running:BuildingWebApplications"by Stoyan Stefanov

"ProReact16"byAdamFreeman

"Fullstack React: The Complete Guide to ReactJS and Friends" by Anthony Accomazzo, Ari Lerner, DavidGuttman,andNate Murray

"React Design Patterns and Best Practices: Build easy to scale modular applications using the most powerfulcomponentsanddesignpatterns" byCarlosSantanaRoldán

DisciplineSpecificElective Basket

CourseContentsforDisciplineSpecificElectives–DSE(ElectiveGroupBasket)

Elective Group	Elective 1 UDIT/DSET/520-523 UDIT/DSEP/520-523	Elective 2 UDIT/DSET/570-573 UDIT/DSEP/570-573	Elective 3 UDIT/DSET/620-623 UDIT/DSEP/620-623	Elective 4 UDIT/DSET/670-673 UDIT/DSEP/670-673
Pattern Analysis & Machine Intelligence	Soft Computing	Fuzzy Systems : Theory, Application & Case Study	Video Processing	Pattern Recognition
Remote Sensing and Geospatial Technology	Fundamental of Satellite Remote Sensing	GIS	Remote Sensing And Digital Image Analysis	Hyperspectral Image Analysis
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 tobefollowedat thetimeofselectionofthecourse.

ElectiveGroup–PatternAnalysisandMachineIntelligence

CourseTitle	**Soft Computing**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 520 UDIT/DSEP/520	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basic knowledge of mathematics and probabilitytheory
- Understandingofcomputerprogrammingconcepts
- Familiaritywithalgorithmsanddatastructures
- Basicunderstandingofartificialintelligenceandmachinelearning

CourseObjectives(CO):

- Tointroducestudentstothefundamentalconceptsandtechniquesofsoftcomputing
- Toprovideacomprehensiveunderstandingoffuzzylogicsystems,neuralnetworks,andevolutionarycomputation
- Toexploretheintegrationofdifferentsoftcomputingmethodologies
- Toequipstudentswiththeskillstoapplysoftcomputingtechniquestoreal-worldproblems
- Toanalyzeandimplementcasestudiesthatdemonstratetheeffectivenessofsoftcomputingapproaches

LearningOutcome(LO):

Bytheend ofthiscourse,studentswillbeableto:

- Understandtheprinciplesandtheoriesunderlyingsoftcomputingtechniques
- Applyfuzzylogicsystems,neuralnetworks,andevolutionarycomputationtosolvecomplexproblems
- Analyzeandevaluatetheperformanceofsoftcomputing models
- Designanddevelopsoftcomputing-basedsolutionsforreal-worldcasestudies

Unit 1: Introduction to Soft Computing - Introduction to soft computing and its significance, Comparison withtraditionalcomputingparadigms,Componentsofsoftcomputing:Neuralnetworks,fuzzylogic,andevolutionarycomputation, Soft computing in real-world applications. Neural Networks - Introduction to artificial neuralnetworks (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: ImagerecognitionusingANNs

Unit 2: Fuzzy Logic - Introduction to fuzzy logic and fuzzy sets, Fuzzy rules and linguistic variables, Fuzzyinferencesystemsandrule-basedreasoning,Fuzzycontrolsystemsandapplications Casestudy:Fuzzylogic-

based temperature control system. Evolutionary Computation - Introduction to evolutionary computation andgenetic algorithms, Representation schemes and fitness evaluation, Selection, crossover, and mutation operators,Geneticprogrammingandapplications Case study:Optimizationusing genetic algorithms

Unit3:HybridApproachesandApplications- Hybridizationofsoftcomputingtechniques,Neuro-fuzzysystemsand their applications, Evolutionary neural networks, Soft computing in data mining, pattern recognition, anddecision supportsystems, Case study:Hybridapproachforstockmarketprediction

RecommendedReferenceBooks:

"SoftComputing:TechniquesandApplications"byS.N.SivanandamandS.N.Deepa"NeuralNetworksandLearning

Machines"bySimon Haykin

"FuzzyLogicwithEngineeringApplications"byTimothyJ.Ross

"Introduction toEvolutionaryComputing"byA. E.Eiben and J.E.Smith

"HybridIntelligentSystems:AnalysisandDesign"bySiddharthaBhattacharyyaandParamarthaDutta

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

Prerequisites:

- Basicknowledgeofmathematics,includingsettheoryandcalculus.
- Familiaritywithbasicconceptsoflogicandreasoning.
- Basicprogrammingskills(preferablyinalanguagesuchasPythonorMATLAB).

CourseObjectives(CO):

- Understandthetheoreticalfoundationsoffuzzysystemsand theirapplications.
- Gainknowledgeoffuzzylogic,fuzzysets,andfuzzyinferencesystems.
- Learntechniquesforfuzzy systemmodeling,control,and decision-making.
- Developtheabilityto designand implementfuzzysystemsforreal-world problems.
- Analyzeand evaluatecasestudiestounderstand thepracticalimplicationsoffuzzysystems.

LearningOutcomes(LO):

Bytheendofthecourse,studentswillbeableto:

- Explaintheprinciplesandconceptsunderlyingfuzzysystems.
- Designandimplementfuzzyinferencesystemsforvariousapplications.
- Applyfuzzymodelingtechniquestodevelopsolutionsforcomplexproblems.
- Evaluateand analyzetheperformanceoffuzzy systemsin real-world scenarios.
- Applyfuzzydecision-makingmethodstosupportcomplexdecisionprocesses.

CourseOutline:

Unit1:IntroductiontoFuzzySystems-Introductiontouncertaintyandimprecision,Fuzzysetsandmembershipfunctions,Fuzzyoperationsandlinguisticvariables,Fuzzyrules andfuzzyreasoning.FuzzySystemsDesignandImplementation - 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 andcontrol in process industries, Fuzzy systems in intelligent transportation systems, Fuzzy systems for patternrecognition. Fuzzy Systems in Finance and Economics - Fuzzy logic in portfolio optimization, Fuzzy modelingforriskassessment, Fuzzytime seriesforecasting,Fuzzy systemsincreditscoring

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

RecommendedReferenceBooks:

"FuzzyLogicwithEngineeringApplications"byTimothyJ.Ross

"Fuzzy Setsand Fuzzy Logic:TheoryandApplications"byGeorgeJ.KlirandBoYuan

"IntroductiontoFuzzySets,FuzzyLogic,andFuzzyControlSystems"byGuanrongChenandTrungTatPham"Fuzzy

SystemsEngineering:Theoryand Practice"byC.L.Philip Chen and Han-PangHuang

"FuzzyLogic:Intelligence,Control,andInformation"byJohnYenandRezaLangari

CourseTitle	**Video Processing**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 620 UDIT/DSEP/620	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- ProficiencyinPythonprogramminglanguage
- Basicunderstandingofimageprocessingconcepts
- FamiliaritywithfundamentalconceptsofcomputerVision

CourseObjectives(CO):

- Understand thefundamentalprinciplesand techniquesofvideoprocessing.
- GainproficiencyinusingPythonlibrariesandframeworksforvideomanipulationandanalysis.
- Applyvariousvideoprocessingtechniquestoextractmeaningfulinformationfromvideostreams.
- Acquiretheskillsto enhanceandrestorevideoqualityusingadvanced algorithms.
- Exploretheapplicationofdeep learningforvideoanalysisand synthesis.
- Developtheabilitytodesign andimplementvideoprocessing solutionsforreal-worldproblems.

Learning Outcomes(LO):

Bytheend ofthiscourse,studentswillbeableto:

- Understandtheunderlyingconceptsandtechniquesofvideoprocessing.
- ManipulateandpreprocessvideodatausingPython.
- Implementmotiondetection,objectrecognition,andfacedetectionalgorithms.
- Enhancevideoqualitythroughdenoising,deblurring,andcolorcorrection.
- Applyvideocompressiontechniquesforefficientstorageandtransmission.
- Utilizedeeplearningmodelsforvideoanalysis,classification,andsynthesis.
- Designanddevelopvideoprocessingsolutionsforreal-worldapplications.

CourseOutline:

Unit 1: Introduction to Video Processing - Fundamentals of video representation and formats, Video acquisitionandpreprocessingtechniques.VideoAnalysisTechniques-Motiondetectionandtracking,Objectrecognitionandtracking, Face detectionandrecognition

Unit 2: Video Enhancement and Restoration - Noise reduction and image stabilization, Video denoising anddeblurring, Contrast enhancement and color correction. Video Compression and Encoding, Principles of videocompression,Videoencodingalgorithmsandstandards,Codecsandformatsforefficientstorageandtransmission

Unit3:DeepLearningfor VideoProcessing-Convolutionalneuralnetworksforvideoanalysis,Videoclassification and action recognition, Video generation and synthesis. Case Studies a) Real-world applications ofvideoprocessingusingPython,b)Casestudiesonsurveillancesystems,videoanalytics,etc.c)Hands-onprojectsto reinforce conceptslearned.

RecommendedReferenceBook:

"PythonforVideoProcessing:TechniquesandCaseStudies"

"DigitalVideoProcessing", SecondEditionbyA. MuratTekalp,June2015,Pearson,ISBN: 9780133991116

CourseTitle	**Pattern Recognition**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 670 UDIT/DSEP/670	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- BasicknowledgeofPythonprogramminglanguage
- FamiliaritywithfundamentalconceptsinmachinelearningUnderstandingoflinearalgebraandprobabilitytheory

CourseObjectives(CO):

- Tounderstandthefundamentalprinciplesandtechniquesofpatternclassification
- Todevelopproficiencyinimplementingpatternclassificationalgorithmsusing Python
- Togainpracticalexperiencethroughhands-onexercisesandreal-worldcasestudies
- Tolearnhowtoevaluateandselectappropriateclassificationmodelsfordifferenttasks
- Toexploretheethicalconsiderationsandchallengesinpatternclassification

LearningOutcome(LO):

Bytheend ofthiscourse,studentswillbeableto:

- Understandtheconceptsandprinciplesofpatternclassification
- ApplyvariouspatternclassificationalgorithmsusingPython
- Preprocessand analyzedatasetsforpattern classificationtasks
- Evaluateandcomparetheperformanceofclassificationmodels
- Implementpatternclassification solutionstoreal-worldproblems

CourseOutline:

Unit1:IntroductiontoPatternClassification -Overviewofpatternclassification,Importanceandapplicationsofpattern classification, Key concepts and terminology. Data Preprocessing and Feature Selection - Data cleaningandpreprocessingtechniques,Featureextractionanddimensionalityreductionmethods,Featureselectionalgorithms

Unit 2: Supervised Learning Algorithms - Linear classifiers (e.g., logistic regression, support vector machines),Decisiontreesandensemblemethods(e.g.,randomforests,boosting),Neuralnetworksanddeeplearningmodels.Unsupervised Learning Algorithms - Clustering techniques (e.g., k-means, hierarchical clustering), DensityestimationandGaussianmixturemodels,Dimensionalityreductionmethods(e.g.,principalcomponentanalysis)

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 withPython-OverviewofPythonlibrariesforpatternclassification(e.g.,scikit-learn,TensorFlow,Keras),Implementingclassification algorithmsusing Python,Handling largedatasetsand optimizing performance.

RecommendedReferenceBook:

"PatternClassificationandMachineLearning",ChristopherM.Bishop,Springer,2006"PatternClassification",Richard O.Duda,Peter E.Hart,David G.Stork,Wiley

Elective Group–Remote Sensing and GeoSpatial Technology

CourseTitle	**Fundamental of Satellite Remote Sensing**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 521 UDIT/DSEP/521	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisite:

Students should have a basic understanding of geographic information systems (GIS) and remote sensingprinciples. Familiarity with computer programming and image processing software would be advantageous butnotmandatory.

CourseObjectives(CO):

- Tointroducestudentstotheprinciplesandtechniquesofsatelliteremotesensing
- Todevelopstudents'skillsinanalyzingandinterpretingsatelliteimagery
- Toprovidehands-onexperiencewithremotesensingsoftwareandtools
- Toexploretheapplicationsofsatelliteremotesensinginvariousdomains
- Toenhancestudents'abilitytocriticallyevaluateandutilizesatellitedataforreal-worldcasestudies

LearningOutcomes(LO):

Bytheend ofthiscourse,studentswillbeableto:

- Understandthefundamentalconceptsandprinciplesofsatelliteremotesensing
- Acquire,preprocess,andenhancesatelliteimageryforanalysis
- Performimageclassificationandinterpretationusingappropriatetechniques
- Applysatelliteremotesensingforlandcovermappingandchangedetection
- Applysatelliteremotesensingforspecificapplications,suchasagriculture,environment,anddisasterassessment
- Criticallyanalyzeand evaluateremotesensing dataforcasestudies

CourseOutline:

Unit 1: Introduction to Remote Sensing - Overview of remote sensing principles and technologies, Types ofremotesensingplatforms,Electromagneticspectrumanditsinteractionwith Earth'ssurface,SatelliteSystemsandSensors - Overview of satellite systems and orbits, Types of satellite sensors and their characteristics, Sensorresolution andimage interpretation

Unit 2: Image Acquisition and Preprocessing - Geometric and radiometric corrections, Image enhancementtechniques, Image fusion and multi-temporal analysis. Image Classification and Interpretation - Supervised andunsupervised classification methods, Object-based imageanalysis,Land coverand land usemapping

Unit 3: Digital Elevation Models (DEMs) and Topographic Analysis - Generation and applications of DEMs,Terrain modeling and slope analysis, Hydrological applications. Change Detection and Time Series Analysis - Techniquesfordetectingandmonitoringchanges, Analysisoftemporalsatellitedata,Urbangrowthanddeforestationstudies.CaseStudies& ApplicationsofSatelliteRemoteSensinga) Agricultureandcropmonitoring,b)Environmentalmonitoringandnaturalresourcemanagement,c)Disasterassessmentandmitigation, d)Urbanplanningandinfrastructuredevelopment

RecommendedReferenceBook:

"RemoteSensingoftheEnvironment:AnEarthResourcePerspective" byJohnR. Jensen

"RemoteSensingandImageInterpretation"byThomasLillesand,RalphW.Kiefer,andJonathanChipman

"Introduction to Satellite Remote Sensing: Atmosphere, Ocean, Land and Cryosphere Applications" by William Emery,Sr. andJoelSusskind

CourseTitle	**Geographic Information System (GIS)**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 571 UDIT/DSEP/571	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisite:

- BasicknowledgeofGeographicInformationSystems(GIS)
- FamiliaritywithspatialdataconceptsandformatsProficiencyinusingGISsoftware (e.g.,ArcGIS,QGIS)

CourseObjectives(CO):

- DevelopadeepunderstandingofadvancedGISconceptsandtechniques
- Acquirepracticalskillsin spatialanalysisand modeling
- Enhanceproficiencyindatamanagementandintegration
- Exploreadvancedcartographyandvisualizationtechniques
- Gainknowledgeingeospatialmodelingandsimulation
- ApplyadvancedGISmethodstosolvereal-world problems

LearningOutcomes(LO):

Bytheend ofthiscourse,studentswillbeableto:

- Applyadvancedspatialanalysistechniquestosolvecomplexspatialproblems
- Designandimplementadvanceddatamanagementstrategiesin GIS
- Createvisuallycompellingandinformativemapsusingadvancedcartographictechniques
- Developgeospatialmodelsandsimulationstounderstandandpredictspatialphenomena
- EvaluateandselectappropriateGIStoolsandmethodologiesforspecificapplications
- Applycriticalthinking skillsto analyzeand solvereal-worldspatialproblemsusingGIS

CourseOutline:

Unit 1: Introduction to GIS Concepts - Overview of advanced GIS techniques and applications, Spatial analysisand modelling, Data integration and interoperability, Advanced Spatial Analysis Techniques - Spatial statisticsand analysis, Network analysis and routing, Geo-statistics and interpolation, Multicriteria decision analysis(MCDA), Spatial regressionanalysis

Unit 2: Advanced Data Management in GIS - Database design and management, Data quality assessment andimprovement, Data integration and fusion, Spatial data infrastructure (SDI) concepts, Data privacy and securityin GIS

Unit 3: Advanced Cartography and Visualization - Advanced cartographic design principles, Thematic mappingtechniques, 3D visualization and virtual reality (VR) in GIS, Web-based mapping and interactive applications,Data-drivencartographyanddynamicmapping,GeospatialModellingandSimulation-Introductiontogeospatialmodelling, Cellular automata and agent-based modelling, Time-series analysis and forecasting, Risk assessmentand hazardmodelling,Urbangrowthmodelling

RecommendedReferenceBook:

"AdvancedGeographicInformationSystems: SpatialAnalysis andModeling", RobertP.Haining,Wiley,2021

CourseTitle	**Remote Sensing And Digital Image Analysis**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 621 UDIT/DSEP/621	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisite:

- Basicknowledgeofgeography,environmentalscience,orarelatedfield.
- Familiaritywithbasiccomputerskillsandimageprocessingconcepts.

CourseObjectives(CO):

- Toprovidestudentswithasolidunderstandingofremotesensingprinciplesandtechnologies.
- Todeveloppracticalskillsindigitalimageprocessingandanalysistechniques.
- Toexplorethevariousapplicationsofremotesensingindifferentdomains.
- Toenhancecriticalthinkingandproblem-solvingabilitiesthroughcasestudyanalysis.
- Topreparestudentsforcareersinremotesensing,geospatialanalysis,andrelatedfields.

LearningOutcomes(LO):

Bytheendofthecourse,studentswillbeableto:

- Understandthefundamentalprinciplesandtechnologiesofremotesensing.
- Applyimagepre-processingtechniquestoenhanceand correctremotesensingdata.
- Performimageclassification,segmentation,andchangedetectionusingremotesensingimagery.
- Analyzeandinterpretremotesensingdatafordifferentapplicationsanddomains.
- Effectivelyuseremotesensingsoftwareandtoolsforimageanalysis.
- Demonstratecriticalthinkingskillsbyevaluatinganddiscussingcasestudies.

CourseOutline:

Unit 1: Introduction to Remote Sensing - Overview of remote sensing principles and technologies, Types ofremotesensingplatformsandsensors,Introductiontosatelliteandaerialimagery,ImagePre-processingTechniques - Image acquisition and data formats, Radiometric and geometric correction methods, Atmosphericcorrection techniques,Image enhancementandnoise reduction

Unit 2: Image Classification and Segmentation - Supervised and unsupervised classification methods, Featureextractiontechniques,Object-basedimageanalysis(OBIA),Changedetectionandmonitoring-Changedetectiontechniques, Temporal analysis of remote sensing data, Monitoring land cover changes. Hyperspectral ImageAnalysis: Hyperspectral data characteristics, Spectral unmixing, Classification of hyperspectral images. CaseStudies and Applications: Land cover mapping, Vegetation monitoring, Urban growth analysis, Environmentalmonitoring

Unit3:AdvancedImageAnalysisTechniques-
Hyperspectralandmultispectralimageanalysis,Fusionofremotesensingdatasources,Textureanalysisandspatialpatternrecognition,Time-seriesanalysis,ApplicationsofDigitalRemote Sensing, Land cover and land use mapping, Environmental monitoring and assessment, Urban planningand infrastructure management, Agriculture and forestry applications, Disaster management and emergencyresponse.

RecommendedReferenceBook:

"DigitalImageProcessing:RemoteSensingPerspectives",JohnR.Jensen,PrenticeHall,2016

"Remote Sensing and Image Interpretation" Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. ChipmanWiley

CourseTitle	**Hyper Spectral Image Analysis**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 671 UDIT/DSEP/671	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basicknowledgeofremotesensingprinciplesandtechniques
- Familiaritywithimageprocessingfundamentals
- Understandingofstatisticalconcepts

CourseObjectives(CO):

- Gainacomprehensiveunderstandingofhyperspectralimagingprinciplesandtechniques
- Developproficiencyin preprocessinghyperspectraldataforanalysis
- Learnadvancedalgorithmsandmethodsforhyperspectralimageclassificationandunmixing
- Acquireskillsindimensionalityreductionandfeatureextractionfromhyperspectraldata
- Exploreadvancedanalysistechniquesandtheirapplicationsin real-worldcasestudies

LearningOutcomes(LO):

Bytheendofthecourse,studentswillbeableto:

- Describetheprinciplesofhyperspectralimaginganditsapplications
- Performpreprocessingtaskssuchasradiometriccorrectionandnoisereduction
- Applysupervisedandunsupervisedclassificationtechniquestohyperspectraldata
- Conductendmemberextractionandabundanceestimationforunmixing
- Implementdimensionalityreductionmethodstoreducedatacomplexity
- Applyadvancedanalysistechniquesfortargetdetection,sub-pixelmapping,andchangedetection
- Analyzeandinterprethyperspectraldatainthecontextofvariouscasestudies

CourseOutline:

Unit 1: Introduction to Hyperspectral Imaging - Basics of hyperspectral data acquisition and characteristics,Spectral signatures and spectral libraries. Hyperspectral Data Preprocessing - Radiometric and atmosphericcorrection,Noise reduction techniques,Calibrationand geometric correction

Unit 2: Hyperspectral Image Classification-Supervisedandunsupervisedclassificationmethods,Featureextraction and selection, Neural networks and deep learning for classification, Hyperspectral Unmixing - Linearandnonlinearunmixingalgorithms,Endmemberextraction, Abundanceestimationandabundancemaps

Unit 3: Dimensionality Reduction - Principal Component Analysis (PCA), Independent Component Analysis(ICA),Non-negativeMatrixFactorization(NMF). AdvancedHyperspectral AnalysisTechniques-Targetdetection and anomaly detection, Sub-pixel mapping, Change detection and time-series analysis, Case Studies inHyperspectralImage Analysis–a) Applicationsinagriculture,b)environmentalmonitoring,c)mineralexploration,andremote sensing

RecommendedReferenceBook:

Hyperspectral Remote Sensing: Principles and Applications, Antonio Plaza, Jon Atli Benediktsson, JosianeZerubia,Academic Press,2016

ElectiveGroup–Security

CourseTitle	**Network Security**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 522 UDIT/DSEP/522	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basicknowledgeofnetworkingconcepts
- Familiaritywithoperatingsystemsandcomputerarchitecture
- Understandingoffundamentalsecurityprinciples

CourseObjectives(CO):

- Understandthefundamentalprinciplesandconceptsofnetworksecurity.
- Identifyandassessnetwork securityrisksandvulnerabilities.
- Evaluateandimplementappropriatenetworksecuritytechnologiesandprotocols.
- Developincidentresponseplansandperformnetworkforensics.
- Analyzeandapplysecuritymeasuresincloudandmobilenetworkenvironments.
- Analyzeandlearnfromreal-worldnetworksecuritycasestudies.

LearningOutcomes(LO):

By the end of this course,studentswillbeableto:

- Applynetworksecurityprinciplestodesignsecurenetworkinfrastructures.
- Implementandconfigurenetworksecuritytechnologieseffectively.
- Analyzeandrespondto networksecurity incidents.
- Evaluateandapplysecuritymeasuresincloudandmobilenetworks.
- Investigateandpresentcasestudiesofnetworksecuritybreaches.

CourseOutline:

Unit1:IntroductiontoNetworkSecurity-Overviewofnetworksecurityconcepts,Securitygoals:confidentiality,integrity,availability,Threatlandscapeandattackvectors,Riskassessmentandmanagement,NetworkArchitectureandDesign-Securenetworkdesignprinciples,Segmentationandzoning,Defense-in-depthstrategies, Security considerationsfor wirelessandmobile networks

Unit 2: Network Perimeter Security - Firewalls and intrusion detection/prevention systems, Virtual PrivateNetworks (VPNs), Network Address Translation (NAT), Demilitarized Zone (DMZ) design, Secure NetworkProtocolsandServices-SecureSocketLayer/TransportLayerSecurity(SSL/TLS),SecureShell(SSH),DomainNameSystem SecurityExtensions(DNSSEC), NetworkTimeProtocolSecurity(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 securityprotocols (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-worldsecuritybreachesb)Investigatingnetworkattacksandincidentsc)Lessonslearnedfromnotablesecurityincidents

RecommendedReferenceBooks:

"NetworkSecurity:PrivateCommunicationinaPublicWorld"byCharlieKaufman,RadiaPerlman,andMikeSpeciner

"FirewallsandInternetSecurity:RepellingtheWilyHacker"byWilliamR.CheswickandStevenM.Bellovin"Network

SecurityEssentials:Applicationsand Standards"byWilliamStallings

"PracticalPacketAnalysis:UsingWiresharktoSolveReal-WorldNetworkProblems" byChrisSanders

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by DafyddStuttard andMarcusPinto

"Hacking:TheArtofExploitation"byJonErickson

"AppliedCryptography:Protocols,Algorithms, andSourceCodeinC" byBruce Schneier

"SecurityEngineering:AGuidetoBuildingDependable DistributedSystems"byRossJ.Anderson

CourseTitle	**CyberSecurity**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 572 UDIT/DSEP/572	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisite:

Basicunderstandingofcomputernetworksandinformationtechnology.

CourseObjectives(CO):

- Toprovidestudentswithasolidfoundationincybersecurityconceptsandprinciples.
- Todevelopanunderstandingofcommoncyberthreatsandvulnerabilities.
- Tofamiliarizestudentswithvarioustoolsandtechniquesusedincybersecurity.
- Toenablestudentstoassessandmitigateriskstodigitalsystemsanddata.
- Topromoteethicalconsiderationsandlegalcomplianceincybersecuritypractices.

LearningOutcomes(LO):

Bytheendofthecourse,studentswillbeableto:

- Identifyandanalyzecommoncyberthreatsandvulnerabilities.
- Implementnetworksecuritymeasurestoprotectagainstattacks.
- Applycryptographictechniquestoensuredataconfidentialityandintegrity.
- Developandimplementsecuresoftwaredevelopmentpractices.
- Demonstrateincidentresponseandmanagementskills.
- Conductethicalhackingandpenetrationtestingactivities.
- Evaluateandrecommendsecuritymeasuresfordifferentscenarios.

CourseOutline:

Unit 1: Introduction to Cyber Security - Overview of cyber security fundamentals, Importance of cyber securityinmodernsociety,LegalandethicalconsiderationsCyberThreatsandAttacks-Typesofcyberthreats:malware,socialengineering,phishing,etc.Attack vectorsandmethods, Risk assessmentand vulnerability analysis

Unit 2: Cybersecurity Tools - Firewall and intrusion detection systems, Anti-malware software and endpointprotection, Network monitoring and log analysis tools, Encryption and data protection tools, Securing NetworksandSystems-Networksecurityprinciplesandprotocols,Secureconfigurationofoperatingsystemsandapplications,Authenticationand accesscontrolmechanisms, Patch managementandvulnerabilitymitigation

Unit 3: Incident Response and Forensics - Incident response lifecycle, Digital forensics techniques, Evidencecollection and analysis, Post-incident recovery and lessons learned, Case Studies in Cyber Security – a) Analysisofreal-worldcybersecurityincidents,b)Examinationofincidentresponsestrategiesc)Lessonslearnedandbestpractices

RecommendedReferenceBooks:

"TheArtofInvisibility" byKevinMitnick

"Hacking:TheArtofExploitation"byJonErickson

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

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by DafyddStuttard andMarcusPinto

"NetworkSecurityBible"byEricCole

"BlueTeamHandbook:IncidentResponseEdition"byDonMurdoch,Jr.

"DigitalForensicsandIncidentResponse:DevelopinganIncidentResponsePlan"byGerardJohansen

CourseTitle	**Cyber Forensics: Tools, Techniques and Case Studies**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 622 UDIT/DSEP/622	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basicknowledgeofcomputersystemsandnetworks
- Familiaritywithoperatingsystemsandfilesystems
- Understandingofinformationsecurityprinciples

CourseObjectives(CO):

- Understand thefundamentalsofcyberforensicsand itsrolein combating cybercrime.
- Gainproficiencyintheuseoftoolsandtechniquesforacquiringandanalyzingdigitalevidence.
- Developskillsinconductingforensicinvestigationsandincidentresponseprocedures.
- Comprehendthelegalandethicalissuesassociatedwithcyberforensics.
- Exploreadvancedtopicssuchasnetworkforensics,mobileandcloudforensics,andmalwareanalysis.
- Applytheoreticalknowledgetoreal-worldcasestudiesandpracticalscenarios.

LearningOutcome(LO):

Uponcompletionofthecourse, studentswillbeableto:

- Identifyandcollectdigitalevidenceusingforensicallysoundmethods.
- Analyzedigitalartifactstouncoverevidenceofcybercrime.
- Conductnetwork andmobiledeviceforensicsinvestigations.
- Employappropriatetoolsandtechniquesto recover andpreservedata.
- Understandthelegalandethicalconsiderationsincyberforensics.

CourseOutline:

Unit1:IntroductiontoCyberForensics-Understandingcyberforensics:scopeandimportance,Legalandethicalconsiderations in cyber investigations, Roles and responsibilities of a cyber forensic examiner. Digital Evidenceand Forensic Processes, Types of digital evidence, Evidence acquisition and preservation, Forensic imaging andhashing, Chainof custodyanddocumentation

Unit 2: Forensic Tools and Techniques - Forensic imaging tools (e.g., EnCase, FTK, Autopsy), File systemanalysisandrecovery,Networkforensicsandloganalysis,Memoryforensicsandanalysis,MobileandCloud

Forensics,Forensicsformobiledevices(smartphones,tablets),Investigatingcloud-basedplatformsandservices,Extraction andanalysisof mobile appdata

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 anddynamicanalysistechniques,Codedecompilationandreverseengineering.CaseStudiesandPracticalApplication – a) Analyzing a cybercrime case from start to finish, b) Examining digital evidence and conducting forensicanalysis,c)Reportingfindingsand presentingevidence incourt

RecommendedReferenceBooks:

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

Britz"ComputerForensics:InvestigatingDataandImageFiles"byEC-Council

"TheBasicsofDigitalForensics:ThePrimerforGettingStartedinDigitalForensics"byJohnSammons"FileSystemForensicAnalysis"

byBrian Carrier

"MalwareForensics:InvestigatingandAnalyzingMaliciousCode"byCameron H.Malin,etal.

"PracticalPacketAnalysis:UsingWiresharktoSolveReal-WorldNetworkProblems" byChrisSanders

CourseTitle	**Cryptography and Blockchain**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 672 UDIT/DSEP/672	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisite:

- Basicunderstandingofcomputerscienceconceptsandprogramming
- Familiaritywithdatastructuresandalgorithms
- Knowledgeofnetworkingandinternetprotocols

Course Objectives(CO):

- Understandthefundamentalconceptsandprinciplesofcryptographyandblockchaintechnology.
- Analyzeandevaluatedifferentcryptographicalgorithms,protocols,andsystems.
- Gainhands-onexperienceinimplementingandsecuringcryptographicsystems.
- Exploretheinnerworkingsofblockchainnetworksandconsensusmechanisms.
- Developtheskillstodesign anddeploydecentralizedapplicationsusingsmartcontracts.
- Investigatereal-worldusecasesofcryptographyandblockchaininvariousindustries.

LearningOutcome(LO):

Uponcompletionofthiscourse, studentswillbeableto:

- Demonstrateacomprehensiveunderstandingofcryptographicalgorithmsandtheirapplications.
- Evaluatethesecurityofcryptographicsystemsandproposecountermeasures.
- Analyzeanddesignblockchainnetworksforspecificusecases.
- Developanddeploysmartcontractsfordecentralizedapplications.
- Applycryptographyandblockchainprinciplesto addressreal-world challenges.

CourseOutline:

Unit 1: Introduction to Cryptography - Overview of cryptography and its historical significance, Symmetric andasymmetricencryptionalgorithms,Hashfunctionsanddigitalsignatures,Cryptographicprotocolsandapplications.Cryptographic TechniquesandAlgorithms-
Keymanagementanddistribution,Publickeyinfrastructure(PKI),Cryptographicattacksandcountermeasures,Securecommunication protocols(SSL/TLS)

Unit2:IntroductiontoBlockchainTechnology-
Evolutionofblockchainanditsapplications,Distributedledgertechnologyandconsensusmechanisms,Cryptocurrenciesandsmartcontracts,Blockchainplatformsandecosystems, Blockchain Security and Privacy- Blockchain vulnerabilities and attack vectors, Privacy andanonymity in blockchain, Tokenization and non-fungible tokens (NFTs), Securing blockchain networks andtransactions

Unit 3: Future Trends and Challenges - Quantum cryptography and post-quantum encryption, Scalability andinteroperability in blockchain, Regulatory and legal considerations, Ethical implications of cryptography andblockchaintechnology,CaseStudiesinCryptographyandBlockchain–a)Real-worldapplicationsofcryptography in finance, healthcare, and government sectors, b) Success stories and challenges of blockchainadoptioninindustriessuchassupplychain,voting, andidentity management.

RecommendedReferenceBooks:

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

Stallings"MasteringBitcoin:UnlockingDigitalCryptocurrencies"by AndreasM. Antonopoulos"BlockchainBasics:ANon-

TechnicalIntroductionin25Steps" byDaniel Drescher

"BlockchainRevolution:HowtheTechnologyBehindBitcoinandOtherCryptocurrenciesisChangingtheWorld"by DonTapscottandAlexTapscott

"The Age of Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" byPaulVigna andMichaelJ. Casey

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

ElectiveGroup–NaturalLanguageProcessing

CourseTitle	**Linguistic Fundamentals: Understanding Language Structure and Analysis**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 523 UDIT/DSEP/523	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisite:

None. This course is open to all students with an interest in linguistics. Prior knowledge of linguistics is notrequired.

CourseObjectives(CO):

- Todevelopacomprehensive understandingofthefundamentalcomponentsoflanguage.
- Toexaminetheinteractionbetweenlinguisticelementsandtheirimpactonlanguagestructure.
- Tocultivateanalyticalandcriticalthinkingskillsinlinguisticanalysis.
- Toexplorelanguagevariationanditssocialandculturalimplications.
- Toapplylinguistictheoriesandmethodstoreal-worldcasestudies.

LearningOutcomes(LO):

Bytheend ofthis course,studentswillbeableto:

- Identifyanddescribethecoreelements oflanguage,including phonetics,phonology, morphology,syntax,semantics, andpragmatics.
- Analyzeandtranscribespeechsoundsusingphoneticsymbols.
- Conductmorphologicalandsyntacticanalysesoflinguisticdata.
- Interpretandanalyzemeaningatthelexical,sentence,anddiscourselevels.
- Applysociolinguisticprinciplestoexaminelanguagevariationandchange.
- Applylinguistictheoriesandmethodstocasestudies,demonstratingcriticalthinking skills.

CourseOutline:

Unit1:IntroductiontoLinguisticFundamentals-Overviewoflinguisticsasafieldofstudy,Languageasasystemof communication, Fundamental branches of linguistics. Phonetics and Phonology - Introduction to phonetics:articulatory, acoustic, and auditory aspects, Phonemic analysis: phonemes, allophones, and phonological rules,Casestudy:Phonological variationsacrossdifferentdialects

Unit 2: Morphology - Basic units of meaning: morphemes, Word formation processes: affixation, compounding,derivation,etc.Casestudy:Analyzingmorphologicalstructuresindifferentlanguages.Syntax-Sentencestructureandphrasetypes,Syntacticcategoriesandconstituents, Casestudy:Parsing andanalyzingsentencestructures

Unit 3: Semantics and Pragmatics - Meaning and interpretation of words, phrases, and sentences, Pragmaticprinciplesandimplicatures,Casestudy:Pragmaticanalysisofspeechacts,LanguageandSociety-Sociolinguistics: language variationandsocial factors,Language acquisitionandbilingualismCase study:Languagepolicyandplanning

RecommendedReferenceBooks:

"AnIntroductiontoLanguage"byVictoriaFromkin,RobertRodman,andNinaHyams

"Linguistics: AnIntroductiontoLinguistic Theory"by Victoria A.Fromkin,Robert Rodman,andNina

Hyams"TheCambridgeEncyclopedia of Language"byDavidCrystal

"LanguageFiles:MaterialsforanIntroductiontoLanguageandLinguistics"byDepartmentofLinguistics,OhioStateUniversity

"IntroducingMorphology"byRochelleLieber

CourseTitle	**Semantics and Pragmatics**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 573 UDIT/DSEP/573	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basicknowledgeoflinguisticsorarelatedfieldisrecommended butnotrequired.
- Familiaritywithintroductory-levelsyntaxandmorphologyisbeneficial.

CourseObjectives(CO):

- Understandthekeyconceptsand theoriesofsemanticsandpragmatics.
- Developanalyticalskillstoidentifyandanalyzedifferentlayersofmeaninginlanguage.
- Recognizeandevaluatetheroleofcontextinshapinglinguisticinterpretation.
- Applytheoreticalknowledgetoreal-worldcasestudiesandpracticalscenarios.
- Enhancecriticalthinkingandproblem-solvingabilitieswithinthefieldoflinguistics.
- Developeffectivecommunicationskillsthroughclassdiscussionsandpresentations.

LearningOutcomes(LO):

Bytheendofthecourse,studentswillbeableto:

- Demonstrateacomprehensiveunderstandingofthecoreprinciplesofsemanticsandpragmatics.
- Analyzeandinterpretthemeaningofwords,sentences,anddiscourseindifferentcontexts.
- Evaluateanddiscusstheimpactofpragmaticfactorsonlanguageuseandinterpretation.
- Applytheoreticalframeworksto analyzeandsolveproblemsrelatedtosemanticsandpragmatics.
- Constructwell-supportedargumentsandengageinacademicdiscussionsonlanguagemeaning.
- Employcriticalthinkingtoassessandinterpretlinguisticphenomenainvariousdomains.

CourseOutline:

Unit 1: Introduction to Semantics and Pragmatics - Overview of semantics and pragmatics, Distinction betweenmeaning and use, Theoretical frameworks in semantics and pragmatics. Semantic Analysis - Word meaning andlexicalsemantics, Sentencemeaning andcompositionalsemantics,Truth-conditionalsemantics

Unit 2: Pragmatic Analysis - Context and implicature, Speech acts and illocutionary force, Conversationalimplicature and inference, Reference and Presupposition - Reference and referring expressions, Presuppositionand presuppositional analysis,Anaphoraanddeixis

Unit3:MeaningandSociety-Pragmaticsofpolitenessandface-saving,Cross-culturalandinterculturalpragmatics, Genderand languageuse.CaseStudiesandApplicationsa)Analyzingdiscourseand conversationb)Pragmaticsinlegalandpoliticaldiscourse, c)Semantic andpragmatic analysisinadvertising

RecommendedReferenceBooks:

"Semantics: A Coursebook" by James R. Hurford, Brendan Heasley, and Michael B. Smith"PragmaticsandDiscourse:AResource

BookforStudents"byJoan Cutting

"MeaningandRelevance"byDeirdreWilsonandDanSperber"Pragmatics"byStephen C. Levinson

"Semantics: An Introduction to Meaning in Language" by Kate Kearns"Pragmatics:AnIntroduction"byYanHuang

CourseTitle	**NaturalLanguageProcessing**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 623 UDIT/DSEP/623	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basicknowledgeofprogrammingconcepts
- Familiaritywithdatastructuresandalgorithms
- Understandingofprobabilityandstatistics
- Someexposureto machinelearningconceptsisbeneficialbutnotmandatory

CourseObjectives(CO):

- UnderstandthefundamentalconceptsandtechniquesinNaturalLanguageProcessing
- DeveloppracticalskillstobuildandevaluateNLPsystems
- GainknowledgeofvariousNLPapplicationsandcasestudies
- AcquiretheabilitytoanalyzeandsolveNLPproblemseffectively
- ExplorethechallengesandethicalconsiderationsinNLP

LearningOutcomes(LO):

Bytheend ofthecourse,studentsshouldbeableto:

- Explain thekey conceptsandmethodologiesin NLP
- ApplyNLPtechniquestopreprocessand analyzetextdata
- BuildandevaluateNLPmodelsfortaskslikesentimentanalysis,namedentityrecognition,machinetranslation,andquestionanswering
- CriticallyanalyzeandcomparedifferentapproachesandalgorithmsinNLP
- IdentifyandaddressethicalconcernsinNLPapplications

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,languagemodels,andtextgeneration.Evaluationmetrics:Precision,recall,F1score,andperplexity.

Unit2:NLPTechniquesandAlgorithms-Textclassification:NaiveBayes,logisticregression,andsupportvectormachines (SVM), Sentiment analysis: Lexicon-based approaches, machine learning models, and deep learningmodels. Named Entity Recognition (NER): Rule-based systems, conditional random fields (CRF), and neuralnetworks. Sequence labeling: Hidden Markov models (HMM) and Conditional Random Fields (CRF). Wordembeddings:Word2Vec,GloVe, and FastText.

Unit3: AdvancedNLP-Topicmodeling:LatentDirichlet Allocation(LDA)andNon-negativeMatrixFactorization(NMF).Textsummarization:Extractiveandabstractiveapproaches.Machinetranslation:Statisticalmethods,neuralmachinetranslation(NMT),andtransformermodels. Questionanswering:Informationretrieval,document retrieval, and passage ranking. NLP in industry: Case studies of NLP applications in various domains,such ashealthcare,finance, andcustomer support.

RecommendedReferenceBooks:

"Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics,and SpeechRecognition" byDaniel JurafskyandJamesH. Martin.

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

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

Schütze."DeepLearningforNaturalLanguageProcessing"byPalashGoyal,SumitPandey,KaranJain,andKaranKumar.

"TextMiningandAnalysis:PracticalMethods,Examples,andCaseStudiesUsingSAS"byGoutamChakrabortyand Murali Pagolu.

CourseTitle	**AI Chatbot Services and Applications**		
CourseCode(TH) CourseCode(PR)	UDIT/DSET/ 673 UDIT/DSEP/673	Course Type	**Mandatory**
Credits	**4 Credits(2TH+2PR)**	Contact Hours(TH) Contact Hours(PR)	**2 Hrs / Week** **4 Hrs / Week**

Prerequisites:

- Basicunderstandingofprogrammingconcepts
- FamiliaritywithPythonoranotherprogramminglanguage
- Knowledgeoffundamentalmachinelearningconcepts

CourseObjectives(CO):

- Understand theprinciplesandtechnologiesbehindAIchatbotservices.
- Gainhands-onexperienceindesigninganddevelopingchatbotsystems.
- Exploretheapplicationsandpotentialusecasesofchatbottechnology.
- Acquireknowledgeofnaturallanguageprocessing(NLP)techniquesforchatbots.
- Developskillsintraininganddeployingchatbotmodelsusingmachinelearning.

LearningOutcome(LO):

Bytheend ofthiscourse,studentswillbeableto:

- Design andimplementfunctionalchatbotsystemsusingAItechniques.
- ApplyNLPmethodstoprocessandinterpretuserinputin chatbotinteractions.
- Utilizemachinelearningalgorithmstotrainandimprovechatbotmodels.
- Evaluateandchooseappropriatechatbotplatformsandtoolsforspecificapplications.
- Considerethical,legal,and privacyimplicationsin chatbotdevelopment

CourseOutline:

Unit1: Introductionto AI ChatbotServices-Overviewof AIchatbotsandtheirsignificance,Historicaldevelopment of chatbots, Applications and benefits of AI chatbot services, Introduction to natural languageprocessing(NLP)andmachinelearning(ML)inchatbotdevelopment.ChatbotDesignandArchitecture-Chatbotdesign principles and user experience considerations, Conversational flow and dialogue management, Backendarchitectureand integrationwith existing systems,Introduction to chatbotdevelopmentplatformsandtools

Unit2:NaturalLanguageProcessing(NLP)Fundamentals-IntroductiontoNLPanditsroleinchatbotservices,Textpreprocessingandtokenization,Namedentityrecognition(NER)andsentimentanalysis,Wordembeddingsandlanguagemodels,MachineLearningforChatbotDevelopment-Introductiontomachinelearningalgorithmsinchatbotdevelopment,Supervised,unsupervised,andreinforcementlearningtechniques, Trainingdatacollection andannotation,Evaluationmetricsforchatbotperformance

Unit3: BuildingRule-BasedChatbots-Rule-basedchatbotdevelopmentapproach,Designingrule-basedchatbotarchitectures, Implementing intent recognition and entity extraction using rule-based methods, Limitations andchallenges of rule-based chatbots. Introduction to Conversational AI - Understanding conversational AI and itscomponents,Introductiontointentrecognitionanddialoguemanagementsystems,Dialogflowandotherconversational AI platforms, Creating intents, entities, and dialogues in conversational AI platforms. ek 10:AdvancedChatbotFeaturesandTechniques-Multilingualchatbotdevelopment,Emotiondetectionandsentimentanalysis in chatbots, Contextual understanding and maintaining conversation context, Implementing voice-basedchatbotinterfaces

RecommendedReferenceBooks:

"Chatbots:AnIntroductionandEasyGuidetoBuildingYourOwn"byMatthewHarper

"DesigningBots:CreatingConversationalExperiences"byAmirShevat

"BuildingChatbotswithPython:UsingNaturalLanguageProcessingandMachineLearning"bySumitRaj

"ConversationalAIandChatbots:Fundamentals,Applications,andNew Trends"byManojPrasadandAmitavaDasgupta

"AppliedArtificialIntelligence:AHandbookforBusinessLeaders"byMariyaYao,AdelynZhou,andMarleneJia