Engineering 2 - Semester 2

Course Code	Course Title					Course Type
MA2202	Probability and Statistics					BSC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	1	0	40	60	4

Course Objectives

- To understand the concept of random variables and expectation.
- To learn various distributions and their applications.
- To study the properties of convergence of random variables.
- To know the concepts of statistics applicable in estimation and testing.

Course Outcomes

1. Use basic counting techniques (multiplication rule, combinations, permutations) to compute probability and odds.
2. Compute conditional probabilities directly and using Bayes' theorem, and check for independence of events.
3. Set up and work with discrete random variables. In particular, understand the Bernoulli, binomial, geometric and Poisson ,Negative Binomial, Hypergeometric distributions
4. Work with continuous random variables. In particular, know the properties of uniform, normal and exponential distributions.
5. Understand the law of large numbers and the central limit theorem.
6. Understand the difference between probability and likelihood functions, and find the maximum likelihood estimate for a model parameter.

Detailed Contents

UNIT-I

Basic concepts of Probability: Review of Random experiment, Sample space, Mutually exclusive events.Properties based on axiomatic definition of probability. Conditional probability. Independent events.

Random Variables: Definition of random variables. Properties of discrete and continuous random variable. Definition and properties of probability mass function and probability density function. Definition of cumulative distribution function and its properties for discrete and continuous distributions.
Multivariate Distributions: Definition and properties of multivariate distribution (continuous and discrete). Joint probability distributions. Marginal probability distributions. Conditional probability distributions.

UNIT-II

Mathematical Expectation: Concept of mathematical expectation of functions of random variables and their significance.

Discrete Distributions: Properties of various discrete distributions: Binomial, Poisson, Negative Binomial, Geometric, Hypergeometric and Discrete uniform distributions.
Continuous Distributions: Properties of various continuous distributions:Uniform, Exponential, Normal, Gamma distributions.

UNIT-III
Functions of Random Variables: Evaluating probability distribution of functions of random variables using CDF technique. Determination of joint probability distribution of functions of random variables using transformations. Using transformations to evaluate the distribution of functions of random variables.
Moments and Moment Generating Functions: Moments about origin, Central moments. Moment generating functions of random variables and its properties.

UNIT-IV
Covariance and Correlation: Definition and properties of covariance and correlation. Definition of bivariate normal distributions. Properties of its marginal distributions.
Inequalities and Limit Theorems: Chebyshev's inequality, Cauchy Schwarz inequality. Convergence in probability. Central limit theorem.
Ordered Statistics: Probability distributions of ordered statistics and their properties.

UNIT-V
Measures of Central Tendency: Mean median and mode for grouped and ungrouped data. Quartiles, variance and percentiles for given data.
Sampling and Estimation of Parameters: Concepts of sampling and estimation of mean and variance of a distribution form the sample.
Linear Regression: Linear regression for relationship between two variables.
Hypothesis Testing: Formulation of hypothesis and alternate hypothesis. One-sided and two-sided tests. Comparison of means.

Text Books

- Gupta, S.C., Kapoor V.K., Fundamentals of Mathematical Statistics (11th Edition), Sultan Chand & Sons, 2002.
- Ross, S.M., Introduction to Probability and Statistics for Engineers and Scientists (4th Edition), Academic Press, 2011.
- Gupta, A., Groundwork of Mathematical Probability and Statistics (5th Edition), Academic Publishers, 2002.

References

1. Miller, I., Miller, M., John E. Freund's Mathematical Statistics with Applications (7th Edition), Pearson Education, Inc., 2009.
2. Feller, W., An Introduction to Probability Theory and its Applications, Volume 1 (3rd Edition), John Wiley & Sons, Inc., 1967.
3. Feller, W., An Introduction to Probability Theory and its Applications, Volume 2 (2nd Edition), John Wiley & Sons, Inc., 1971.

Course Code	Course Title					Course Type
CE2804	Engineering Graphics					ESC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	1	0	4	40	60	3

Course Objectives

- To introduce the students to the "Universal Language of Engineers" for effective communication through drawing.
- To understand the basic concepts of drawing through modern techniques.
- To impart knowledge about standard principles of projection of objects.
- To provide the visual aspects of Engineering drawing using Auto-CAD.

Course Outcomes

- At the end of the course, the student will be able to
- Use Engineering principles and techniques to understand and interpret engineering drawings.
- Understand the concepts of Auto-CAD.
- Draw orthographic projections of lines, planes and solids using Auto-CAD.
- Use the techniques, skills and modern engineering tools necessary for engineering practices.

Detailed Contents

UNIT-I

Introduction to Engineering Drawing: Principles of Engineering Graphics and their significance, usage of Drawing instruments, lettering, types of lines and Dimensioning.Over view of Auto-CAD: Theory of CAD software (The Menu System, ToolBars, drawing area,Dialogue boxes, Shortcut Menu, the command lines, Select and erase objects, Introduction to layers etc.) Drawing simple figures- lines, planes, solids.

UNIT-II

Geometrical constructions: Construction of regular polygons.Conic sections: Construction of Ellipse, Parabola, Hyperbola (General method only), Cycloid, Epicycloid,Hypocycloid and Involutes.
Scales: Construction of Plain, Diagonal and Vernier scales.

UNIT-III

Orthographic projections: Principles of Orthographic Projections
Projections of Points: Projections of Points placed in different quadrants
Projection of lines: lines parallel and inclined to both the planes (Determination of true lengths and true inclinations and traces) Projection of planes: Planes inclined to both the reference planes.

UNIT-IV

Projection of Solids: Projection of solids whose axis is parallel to one of the reference planes and inclined to the other plane, axis inclined to both the planes Electronics and Communication Engineering Page 42 Projection of sectioned solids: Sectioning of simple solids like prism, pyramid, cylinder and cone in simple vertical position when the cutting plane is inclined to the one of the principal planes and perpendicular to the other – obtaining the true shape of the section.

UNIT-V **Development of surfaces**:Development of surfaces of Right Regular Solids - Prism, Pyramid, Cylinder and Cone Isometric Projections: Principles of Isometric projection – Isometric Scale, Isometric Views of planes and simple solids

Perspective projections: Basic concepts of perspective views.

Text Books

- Bhatt N.D., Panchal V.M. & Ingle P.R., (2014), Engineering Drawing, Charotar Publishing House
- Gopalakrishna K.R., "Engineering Drawing" (Vol. I&II combined), Subhas Stores, Bangalore, 2007.

References

1. Shah, M.B.& Rana B.C. (2008), Engineering Drawing and Computer Graphics, Pearson Education
2. Venugopal K. and Prabhu Raja V., "Engineering Graphics", New Age publications
3. Agrawal B. & Agrawal C. M. (2012), Engineering Graphics, TMH Publication
4. Narayana, K.L. & P Kannaiah (2008), Text book on Engineering Drawing, Scitech Publishers(Corresponding set of) CAD Software Theory and User Manuals

Course Code	Course Title					Course Type
CS2203	Design and Analysis of Algorithms					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- Introduces the notations for analysis of the performance of algorithms.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
- Describes how to evaluate and compare different algorithms using worst-, average-, and best-case analysis.
- Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NPcomplete.

Course Outcomes

1. Students will understand the basic concepts of algorithm design and analysis.
2. Students will be able to analyze and evaluate algorithm performance using asymptotic notations.
3. Students will learn to apply various algorithmic techniques such as divide and conquer, dynamic programming, and greedy method.
4. Students will gain proficiency in solving problems using backtracking and branch and bound methods.
5. Students will gain an understanding of NP-hard and NP-complete problems and their implications in the real world.

Detailed Contents

UNIT-I

Introduction:

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithmic Efficiency –Asymptotic Notations and their properties. Analysis Framework – Empirical analysis – Mathematical analysis for Recursive and Non-recursive algorithms – Visualization

UNIT-II

Brute Force – String Matching , Closest-Pair and Convex-Hull Problems.
Exhaustive Search – Traveling Salesman Problem, Knapsack Problem , Assignment problem.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication,convex hull,closest pair, large integer multiplication.

UNIT-III

Dynamic programming – Principle of optimality,Chain Matrix Multiplication, Computing a Binomial Coefficient , Floyd's algorithm , Multistage graph , Optimal Binary Search Trees , Knapsack Problem and Memory functions.

Greedy Technique –Prim's algorithm and Kruskal's Algorithm , Fractional Knapsack problem, Optimal Merge pattern – Huffman Trees.

UNIT-IV

Backtracking: General method, applications–n-queen problem, sum of subsets problem,graph coloring, Hamiltonian cycles.

Branch and Bound: General method, applications - Traveling Salesperson Problem, 0/1 knapsack problem, Assignment problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

UNIT-V

NP-Hard and NP-Complete problems: NP Hard and NP completeness: Basic concepts, cook's theorem, NP-hard graph problems and scheduling problem, NP- hard code generation problems, Clique Decision problem, Node covering problem, scheduling problem, NP hard code generation problem.

Approximation Algorithms for NP-Hard Problems – Traveling Salesman problem,Knapsack problem.

Text Books

- Horowitz, E., Sahni, S., and Rajasekaran, S. (2019). Fundamentals of Computer Algorithms. University Press.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). Introduction to Algorithms, Second Edition. Pearson Education.

References

1. A. V. Levitin, "Introduction to the Design and Analysis of Algorithms," Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
2. A. Aho, J. Ullman, and M. Hopcroft, "Design and Analysis of Algorithms," Pearson Education.
3. M. T. Goodrich and R. Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples," John Wiley and Sons.

Course Code	Course Title					Course Type
CS2803	Design and Analysis of Algorithms Lab					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	0	0	3	40	60	1.5

Course Objectives

- To develop proficiency in programming algorithms using C/C++.
- To introduce the fundamental concepts of data structures and algorithms.
- To provide practical experience in implementing various data structures and algorithms.
- To understand the techniques used in solving various computational problems.
- To develop problem-solving skills and logical reasoning.

Course Outcomes

At the end of the course the students will have the

1. Ability to write C/C++ programs for various algorithms and data structures.
2. Knowledge of fundamental concepts and terminology related to data structures and algorithms.
3. Proficiency in implementing basic data structures such as linked lists, trees, and graphs.
4. Ability to analyze the time and space complexity of algorithms using asymptotic notations.
5. Problem-solving skills using different algorithmic paradigms such as divide-and-conquer, dynamic programming, greedy algorithms, and backtracking.

List of Experiments:

All the problems have to be implemented either writing C programs or writing C++ programs. Elementary Problems:

1) Using a stack of characters, convert an infix string to a postfix string.
2) Implement polynomial addition using a single linked list
3) Implement insertion, deletion, searching of a BST, Also write a routine to draw the BST horizontally.
4) Implement binary search and linear search in a program
5) Implement heap sort using a max heap.
6) Implement DFS/ BFS routine in a connected graph
7) Implement Dijkstra's shortest path algorithm using BFS
8) Greedy Algorithm (Any Two)
    a) Given a set of weights, form a Huffman tree from the weight and also find oot the code corresponding to each weight.
    b) Take a weighted graph as an input, find out one MST using Kruskal/ prim's algorithm
    c) Given a set of weight and an upper bound M – Find out a solution to the Knapsack problem
9) Divide and Conquer Algorithm (any Two)
    a) Write a quick sort routine, run it for a different input sizes and calculate the time of running. Plot in graph paper input size versus time.

b) Implement two way merge sort and calculate the time of sorting

c) Implement Strassen's matrix multiplication algorithm for matrices whose order is a power of two.

10) Dynamic programming

a. Given two sequences of character, find out their longest common subsequence using dynamic programming

Course Code	Course Title					Course Type
CS2201	Web Technologies					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- To understand the basics of Web Designing using HTML, DHTML, and CSS
- To learn the basics about Client side scripts and Server side scripts

Course Outcomes

At the end of the course the students will have the

1. Ability to design and develop client side scripting techniques
2. Ability to build real world applications using client side and server side scripting languages

Detailed Contents

Unit – I

HTML- List, Tables, Images, Forms, Frames, Cascading Style sheets. XML- Document type definition, XML Schemas, Document Object model

Unit – II

Java Script -Control statements, Functions, Arrays, Objects, Events, Dynamic HTML with Java Script, Ajax

Unit – III

Web servers – IIS (XAMPP, LAMPP) and Tomcat Servers. Java Web Technologies- Servlets, JavaServer Pages, Java Server Faces, Web Technologies in Netbeans, Building a Web Application in Netbeans, JSF Components, Session Tracking, Cookies

Unit – IV

PHP- Basics, String Processing and Regular Expressions, Form Processing and Business Logic, Using Cookies, Dynamic Content, Operator Precedence Chart

Unit – V

Database Connectivity with MySQL - Servlets, JSP, PHP. Case Studies- Student information system,

Health Management System
Text Books • Paul J. Deitel, Harvey M. Deitel, Abbey Deitel, “Internet & World Wide Web How to Program”, Deitel series, 5th edition, 2012 • Jason Gilmore, “Beginning PHP and MySQL From Novice to Professional”, 4th Edition, Apress Publications, 2010
References 1. Robert W. Sebesta, “Programming with World Wide Web”, Pearson, 4th edition, 2008 2. David William Barron, “The World of Scripting Languages”, Wiley Publications, 2000

Course Code	Course Title					Course Type
CS2801	Web Technologies Lab					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	0	0	3	40	60	1.5

Course Objectives

1. To learn the basics in web designing using HTML, CSS, and XML
2. To develop web applications using JSP, Servlets, PHP, and Net Beans

Course Outcomes

At the end of the course the students will have the

1. Ability to design and develop web pages using HTML, CSS, and XML
2. Ability to design and deploy real world applications using client side and server side scripting languages

List of Experiments

- Designing static web pages using HTML
- Designing dynamic web pages using different cascading style sheets
- Designing XML Schemas
- Programs using Java Script
- Programs using Java servlets and JSP
- Designing web applications using PHP
- Designing web applications in Net Beans Environment
- Database Connectivity with MySQL using Java Servlets, JSP, and PHP

Course Code	Course Title					Course Type
CS2202	Operating System					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
- To know the components and management aspects of concurrency management.

Course Outcomes

1. Create processes and threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
3. For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement a file management system.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of
6. a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

Detailed Contents

UNIT-I

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS – Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. System call interface for process management-fork, exit, wait, waitpid, exec

Unit 2:

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multi threads. Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time. Scheduling algorithms: Pre-emptive and Non-preemptive, FCFS, SJF, RR, Priority, Preemptive Priority, SRTF.

Unit 3:

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion,Hardware Solution, Strict Alternation, Peterson's Solution, The Producer-Consumer Problem, Semaphores, Message Passing. Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem etc.Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm(Safety & Resource request algorithm), Deadlock detection and Recovery.

Unit 4:

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition– Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Least Recently used (LRU).

Unit 5:

I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN,LOOK,C-LOOK,Disk reliability, Disk formatting, Boot-block, Bad blocks.

Text Books:

- Silberschatz, A., Galvin, P.B., & Gagne, G. (2013). Operating System Concepts Essentials, 9th Edition. Wiley Asia Student Edition.

References

1. A. Silberschatz, P. Galvin, and G. Gagne, Operating System Concepts Essentials, 9th ed. Wiley Asia Student Edition.
2. W. Stallings, Operating Systems: Internals and Design Principles, 5th ed. Prentice Hall of India.
3. C. Crowley, Operating System: A Design-oriented Approach, 1st ed. Irwin.
4. G. J. Nutt, Operating Systems: A Modern Perspective, 2nd ed. Addison-Wesley.
5. M. Bach, Design of the Unix Operating Systems, 8th ed. Prentice-Hall of India.
6. D. P. Bovet and M. Cesati, Understanding the Linux Kernel, 3rd ed. O'Reilly and Associates.

Course Code	Course Title					Course Type
CS2802	IT Workshop					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	0	0	3	40	60	1.5

Course Objectives

- The objective of this course is to teach students how to use various tools and technologies for scientific computing and data analysis. By the end of the course, students will have gained proficiency in manipulating and analyzing data, creating professional scientific documents, and presenting results visually. The course covers topics such as Unix commands, LaTeX, PHP, NumPy, Matplotlib, Scipy, and statistical analysis in Python using seaborn.

Course Outcomes

This course teaches students to:

1. Use Unix commands for file management and text processing.
2. Create professional-looking scientific documents using LaTeX.
3. Write PHP scripts for web development and connect to MySQL databases.
4. Use NumPy for scientific computing and perform numerical operations on arrays.
5. Create and customize plots using Matplotlib.
6. Process and manipulate images using NumPy and Scipy.
7. Conduct statistical analysis in Python using seaborn for visualization, and hypothesis testing, linear models, and analysis of variance.

Week1

Basic Unix commands

Week2

sed, grep, sort, ssh,awk,shutdown,ftp,service,chown,chmod

Week3

Latex:Introduction, Document Structure,Essentials ,Troubleshooting Creating a Title, Sections ,Labelling,Table of Contents. Typesetting Text: Font Effects, Coloured Text ,Font Sizes ,Lists ,Comments & Spacing,Special Characters.

Week4

Tables, Figures

Equations: Inserting EquationsMathematical Symbols,Practical Inserting References: Introduction,The BibTeX file ,Inserting the bibliography,Citing references,Styles,Practical

Week5

Introduction to PHP Declaring Variables,Data types, Arrays, Strings,Operators,Expressions,Control Structures, Functions, Reading data from Web forms,

Week6

Handling file uploads,Connecting to database (MySQL), Executing Sample Queries, Handling Results, Handling Sessions and Cookiess

Week7

The NumPy array object, What are NumPy and NumPy arrays, Creating arrays, Basic data types, Basic visualization, Indexing and slicing, Copies and views, Fancy indexing, Numerical operations on arrays Elementwise operations, Basic reductions, Broadcasting, Array shape manipulation, Sorting data

Week8

Matplotlib: plotting

Introduction, Simple plot, Figures, Subplots, Axes and Ticks, different types of Plots: examples and exercises

Week9

Image manipulation and processing using Numpy and Scipy

Opening and writing to image files,Displaying images, Basic manipulations- Statistical information, Geometrical transformations, Image filtering – Blurring/smoothing, Sharpening, Denoising, Mathematical morphology, Feature extraction- Edge detection, Segmentation

Week10

High-level scientific computing

File input/output, Special functions, Linear algebra operations, Interpolation: scipy.interpolate, Optimization and fit, Statistics and random numbers

Week 11

High-level scientific computing

Numerical integration, Fast Fourier transforms, Signal processing, Image manipulation

Week12

Statistics in Python, Data representation and interaction, Hypothesis testing: comparing two groups,Linear models, multiple factors, and analysis of variance, More visualization: seaborn for statistical exploration

References

- J. Peek, G. Todino, and J. Strang, "Learning the Unix Operating System," 5th ed., Sebastopol, CA, USA: O'Reilly Media, Inc., 2001.
- L. Lamport, "LaTeX: A Document Preparation System," 2nd ed., Reading, MA, USA: Addison-Wesley Professional, 1994.
- L. Ullman, "PHP and MySQL for Dynamic Web Sites," 5th ed., Berkeley, CA, USA: Peachpit Press, 2017.

- W. McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython," 2nd ed., Sebastopol, CA, USA: O'Reilly Media, Inc., 2017.

Engineering 3 - Semester 2

Course Code	Course Title					Course Type
HS3203	Soft Skills					HSMC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	0	0	2			1

Course Objectives

- To enable students to speak effectively in formal and informal situations.
- To equip the students with necessary writing skills in order to face the corporate world.
- To strengthen the writing skills of the students and help them in documentation.
- To enable students to sharpen their communication skills towards writing a persuasive resume and effective job application letters.
- To equip students with pre-presentation steps, to understand the structure of a good presentation, and devise various techniques for delivering a successful presentation.
- To make students understand the importance of teamwork and group presentations and group discussions.

Course Outcomes

Students will be able

1. Communicate effectively in formal and informal situations.
2. Understand the structure and mechanics of writing resumes, reports, documents and E-mails.
3. Present effectively in academic and professional contexts.
4. Develop communication in writing for a variety of purposes.
5. Identify areas of evaluation in Group Discussions conducted by organizations as part of the selection procedure.
6. Overcome stage fear and tackle questions.

Detailed Contents

UNIT-I

Activities on Fundamentals of Interpersonal CommunicationStarting a conversation - responding appropriately and relevantly - using the right body language-Role Play in different situations & Discourse Skills using visuals.

UNIT-II

Activities on Reading Comprehension General Vs Local comprehension- reading for facts- guessing meanings from context- scanning- skimming- inferring meaning- critical reading – surfing Internet

UNIT-III

Activities on Writing Skills Structure and presentation of different types of writing- Resume writing/ e-

correspondence/ Technical report writing- planning for writing - improving one's writing.

UNIT-IV

Activities on Presentation Skills Oral presentations (individual and group) through JAM sessions/seminars/PPTs and written presentations

UNIT-V

Activities on Group Discussion, Debate and Interview Skills - Dynamics of group discussion-intervention- summarizing-modulation of voice-body language-relevance-fluency and organization of ideas and rubrics for evaluation- Concept and process-pre-interview planning-opening strategies-answering strategies- interview through tele-conference & video-conferencing - Mock Interviews

Text Books

- Soft Skills Training: A workbook to Develop Skills for Employment – By Frederick H.

References

1. Wentz Everyone Communicates, Few People Connect: What the Most Effective.
2. People do Differently – By John C. Maxwell
3. How to Talk to Anyone: 92 Little Tricks to Have Big success in Relationships – By Leil Lowndes.
4. Teamwork101: What Every Leader Needs to Know – By John C. Maxwell.
5. AdaptAbility: How to Survive Change You Didn't Ask For- By M.J. Ryan.
6. Conflict Communication: A New Paradigm in Conscious Communication – By Rory Miller.

Course Code	Course Title					Course Type
EC3203	Introduction to Internet of Things					ESC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- Students will understand the concepts of Internet of Things and be able to build IoT applications.

Course Outcomes

1. The Student is expected to design and develop an IoT real-world application in a specific domain armed with knowledge of Python and choosing hardware for specific application.

Detailed Contents

UNIT-I

Introduction & Concepts Introduction & Concepts: Introduction to Internet of Things- Definitions & Characteristics of IoT, Physical Design of IOT-Things in IoT, IoT Protocols, Logical Design of IOT-IoT Functional Blocks, IoT Communication Models, IoT Communication APIs, IOT Enabling Technologies-Wireless Sensor Networks, Cloud Computing, Big Data Analytics,Communication Protocols, Embedded Systems, IOT Levels & Deployment Templates.

UNIT-II

Domain Specific IoTs and M2M IoT applications for Home Automation, Cities, Environment, Energy,Retail, Logistics, Agriculture, Industry, health and Lifestyle IoT and M2M – Introduction to M2M, Similarities and Differences between IoT and M2M.

UNIT-III

IoT Platforms Design Methodology Introduction, IoT Design Methodology Steps-Purpose and Requirements Specification, Process Specification, Domain Model Specification, Information Model Specification, Service Specifications, IoT Level Specification, Functional View Specification, Operational View R22, Electronics and Communications Engineering Page 165 Specification, Device and Component Integration, Application Development, Case Study on IoT System for Weather Monitoring.

UNIT-IV

Introduction to Python Motivation for using Python for designing IoT systems, Language features of Python, Data types- Numbers, Strings, Lists, Tuples, Dictionaries, Type Conversions, Data Structures: Control of flow-if, for, while, range, break/continue, pass, functions, modules, packaging, file handling, data/time operations, classes, Exception handling, Python packages of Interest for IoT - JSON, XML, HTTPLib, URLLib, SMTPLib

UNIT-V

IoT Physical Devices and End Points Basic building blocks of an IoT device, Rasberry Pi-About theRasberry Pi board, Rasberry Pi interfaces-Serial, SPI,I2C, Interfacing an LED and switch with RPi and controlling. Other IoT Devices- pcDuino, BeagleBone Black, CubieboardIoT

Text Books

- Misra, S., Mukherjee, A., & Roy, A. (2022). Introduction to IoT. Wiley.

References

1. V. Madisetti and A. Bahga, "Internet of Things (A Hands-on-Approach)," 1st ed., VPT, USA: VPT, 2014.
2. F. daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything," 1st ed., Apress, USA: Apress Publications, 2013.

Course Code	Course Title					Course Type
CS3201	DevOps					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	60	40	3

Course Objectives

- Understand the skill sets and high-functioning teams involved in Agile, DevOps and related
- methods to reach a continuous delivery capability.
- Implement automated system update and DevOps lifecycle

Course Outcomes

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

1. Understand the various components of the DevOps environment.
2. Identify Software development models and architectures of DevOps
3. Use different project management and integration tools.
4. Select an appropriate testing tool and deployment model for project.

UNIT-I

Introduction to DevOps:

Introduction, Agile development model, DevOps and ITIL. DevOps process and Continuous Delivery, Release management, Scrum, Kanban, delivery pipeline, identifying bottlenecks.

UNIT-II

Software development models and DevOps:

DevOps Lifecycle for Business Agility, DevOps, and Continuous Testing. DevOps influence on Architecture: Introducing software architecture, The monolithic scenario, Architecture rules of thumb, The separation of concerns, Handling database migrations, Micro services and the data tier, DevOps, architecture, and resilience.

UNIT-III

Introduction to project management:

The need for source code control, the history of source code management, Roles and code, source code management system and migrations, shared authentication, Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, GitLab.

UNIT-IV

Integrating the system:

Build systems, Jenkins build server, Managing build dependencies, Jenkins plugins, and file system layout, The host server, Build slaves, Software on the host, Triggers, Job chaining and build pipelines, Build servers and infrastructure as code, Building by dependency order, Build phases, Alternative build servers, Collating quality measures.

UNIT-V

Testing Tools and Deployment:

Various types of testing, Automation of testing Pros and cons, Selenium - Introduction, Selenium features, JavaScript testing, Testing backend integration points, Test-driven development, REPL-driven development. Deployment of the system: Deployment systems, Virtualization stacks, code execution at the client, Puppet master and agents, Ansible, Deployment tools: Chef, Salt Stack and Docker

Text Books

- Joakim Verona., Practical DevOps, Packt Publishing, 2016

References

1. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley
2. publications.
3. 2. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison
4. Wesley

Course Code	Course Title					Course Type
CS3801	DevOps Lab					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	0	0	3	40	60	1.5

Course Objectives

- Develop a sustainable infrastructure for applications and ensure high scalability. DevOps aims to shorten the software development lifecycle to provide continuous delivery with high-quality.

Course Outcomes

1. Understand the need of DevOps tools
2. Understand the environment for a software application development
3. Apply different project management, integration and development tools
4. Use Selenium tool for automated testing of application

List of experiments:

1. Write code for a simple user registration form for an event.
2. Explore Git and GitHub commands.
3. Practice Source code management on GitHub. Experiment with the source code in exercise 1.
4. Jenkins installation and setup, explore the environment.
5. Demonstrate continuous integration and development using Jenkins.
6. Explore Docker commands for content management.
7. Develop a simple containerized application using Docker.
8. Integrate Kubernetes and Docker
9. Automate the process of running containerized application for exercise 7 using Kubernetes.
10. Install and Explore Selenium for automated testing.
11. Write a simple program in JavaScript and perform testing using Selenium.
12. Develop test cases for the above containerized application using selenium.

Course Code	Course Title					Course Type
CS3202	Machine Learning					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
Knowledge on statistical methods	3	0	0	40	60	3

Course Objectives

- To provide an introduction to the field of machine learning and its applications
- To teach the fundamental concepts of linear regression, decision trees, instance-based learning, and clustering
- To enable students to understand the basics of artificial neural networks and support vector machines
- To introduce students to ensemble learning techniques and recommender systems
- To provide hands-on experience in implementing machine learning algorithms using Python

Course Outcomes

1. Understand the basic concepts of machine learning and its applications
2. Apply linear regression and decision tree learning techniques to real-world problems
3. Understand and apply instance-based learning and clustering algorithms
4. Implement artificial neural networks and support vector machines
5. Apply ensemble learning techniques and develop recommender systems
6. Use Python to implement machine learning algorithms and evaluate their performance

Detailed Contents

UNIT I

Introduction: Introduction to Machine Learning: Introduction. Different types of learning, Hypothesis space and inductive bias, Evaluation. Training and test sets, cross validation, Concept of over fitting, under fitting, Bias and Variance.

Linear Regression: Introduction, Linear regression, Simple and Multiple Linear regression, Polynomial regression, evaluating regression fit.

UNIT II

Decision tree learning: Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning, Python exercise on Decision Tree.

Instance based Learning: K nearest neighbor, the Curse of Dimensionality, Feature Selection: forward search, backward search, univariate , multivariate feature selection approach, Feature reduction (Principal Component Analysis) , Python exercise on kNN and PCA.

Recommender System: Content based system, Collaborative filtering based.

UNIT III

Probability and Bayes Learning: Bayesian Learning, Naïve Bayes, Python exercise on Naïve Bayes, Logistic Regression.

Support Vector Machine: Introduction, the Dual formulation, Maximum margin with noise, nonlinear SVM and Kernel function, solution to dual problem.

UNIT IV

Artificial Neural Networks: Introduction, Biological motivation, ANN representation, appropriate problem for ANN learning, Perceptron, multilayer networks and the back propagation algorithm,

UNIT V

Ensembles: Introduction, Bagging and boosting, Random forest, Discussion on some research papers.

Clustering: Introduction, K-mean clustering, agglomerative hierarchical clustering, Python exercise on k-mean clustering.

Text Books

- Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.

References

1. T. Mitchell, "Machine Learning," First Edition, McGraw-Hill, 1997.
2. E. Alpaydin, "Introduction to Machine Learning," MIT Press, 2020.
3. C. Bishop, "Pattern Recognition and Machine Learning," Springer, 2007.

Course Code	Course Title					Course Type
CS3802	Machine Learning Lab					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	0	0	3	40	60	1.5

Course Objectives

- The objective of this lab is to get an overview of the various machine learning techniques and can able to demonstrate them using python.

Course Outcomes

After the completion of the course the student can able to:

1. Understand complexity of Machine Learning algorithms and their limitations;
2. Understand modern notions in data analysis-oriented computing;
3. Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
4. Be capable of performing experiments in Machine Learning using real-world data.

List of Experiments

1. Basic exercises on Python Machine Learning Packages such as Numpy, Pandas and matplotlib.
2. Given a dataset. Write a program to compute the Covariance, Correlation between a pair of attributes. Extend the program to compute the Covariance Matrix and Correlation Matrix.
3. Given a set of sample points in N dimensional feature space. Write a program to fit the points with a hyper plane using Linear Regression. Calculate sum of residual error.
4. Write a program that provides option to compute different distance measures between two points in the N dimensional feature space. Consider some sample datasets for computing distances among sample points.
5. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
6. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
7. Write a program to implement feature reduction using Principle Component Analysis
8. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
9. Given a dataset for classification task. Write a program to implement Support Vector Machine and estimate its test performance.
10. Write a program to implement perceptron for different learning tasks.
11. Write programs to implement ADALINE and MADALINE for a given learning task.
12. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
13. Write a program to implement K means clustering algorithm. Select your own dataset to test the program. Demonstrate the nature of output with varying value of K.

Course Code	Course Title					Course Type
CS3203	Automata Theory and Compiler Design					PCC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	1	0	40	60	4

Course Objectives

- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- To understand deterministic and non-deterministic machines and the differences between decidability and undecidability.
- Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
- Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, intermediate code generation

Course Outcomes

1. Able to employ finite state machines for modeling and solving computing problems.
2. Able to design context free grammars for formal languages.
3. Able to distinguish between decidability and undecidability.
4. Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
5. Acquire skills in using lex tool and design LR parsers

Detailed Contents

UNIT-I

Introduction: Alphabet, Languages and grammars, productions and derivations, Chomsky hierarchy of languages. Regular Languages and finite automaton: Regular Expressions and languages, deterministic finite automaton and Equivalence with regular expressions, Non deterministic finite automaton (NFA) and Equivalence with DFA, minimization of finite automata.

Regular languages: regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages.

UNIT-II

Context free languages: Context free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, pumping lemma for context free languages, parse trees, ambiguity in CFG, closure properties of CFLs.

Pushdown automata: Deterministic Push down automata (PDA), non deterministic Push down automata (PDA) Context sensitive languages, Context sensitive grammars.

UNIT-III

Turing Machine: Basic model for TM, Turing recognizable (recursively enumerable) and Turing-decidable(recursive) languages and their closure properties, variants of Turing Machines, Non deterministic TMs .

Undicidability: Church-Turing thesis, universal turing machine, the universal and diagnolization languages, undecidable problems about languages.

Introduction – Language Processors, Structure of a compiler- phases of compiler design and overview. Applications of compiler Technology.

Lexical analysis: The role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical errors, error recovery in lexical analysis phase, The Lexical-Analyzer Generator Lex.

UNIT-IV

Syntax Analysis –Top-Down parsing: Brute Forcing, Recursive Descent parsing, LL (1) parsing, Bottom-Up parsing : Shift reduce parsing, conflicts during shift reduce parsing, Introduction to LR Parsing: LR(0), simple LR, powerful LR parsers: CLR, LALR, conflicts, Parser Generators – Yacc. Error Recovery: Introduction, Error detecting and Reporting, Syntax Errors handling.

UNIT-V

Semantic Analysis – Introduction, semantic errors, attribute grammars

Syntax Directed Translation – Syntax Directed Definitions, Evaluation Orders for SDDs. Applications of Syntax Directed Translation. Symbol Table Organization

Intermediate code generation – Variants of syntax trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking.

Run time Environment – storage organization, Stack allocation of space, activation records, and Access to non local data.

Code Generation – Issues in the Design of a Code Generator, the Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks. Peephole Optimization, Register Allocation and Assignment, Instruction Scheduling. Machine Independent Optimizations – The Principal Sources of Optimizations.

Text Books

- J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson education Asia.
- K. D. Cooper and L. Torczon, Engineering a Compiler, Morgan Kaufman, 2012.

References

1. A.V. Aho, Monica Lam, Ravi Sethi, and J.D. Ullman, "Compilers: Principles, Techniques, and Tools," 2nd ed., Addison-Wesley, 2007.
2. Harry R Lewis and Christos H Papadimitriou, "Elements of the Theory of Computation," Pearson education Asia.
3. Dexter C Kozen, "Automata and Computability," Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, "Introduction to the Theory of Computation," PWS publishing.
5. John Martin, "Introduction to Languages and the Theory of Computation," McGraw-Hill Education, 2003.
6. K.C. Louden, "Compiler Construction: Principles and Practice," Cengage Learning, 1997.
7. D. Brown, J. Levine, and T. Mason, "LEX and YACC," O'Reilly Media, 1992.

Course Code	Course Title					Course Type
CS3221	Natural Computing					PEC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- Provide students an insight to the way Nature's computation happens to evolve for betterment. Various natural processes will be explored by which Nature solves problems and optimizes itself. This knowledge is expected to inspire the learners and research for solving various engineering problems, especially the ones dealing with computations.
- Along with Nature inspired techniques for problem solving, the course throws light on various computational aspects which help in simulation and emulation of Nature and its elements. This helps in understanding and analyzing Natural phenomenon.
- Finally, the course concludes by exploring how natural material like DNA and atomic and sub-atomic elements provide distinguished way of solving computational problems.

Course Outcomes

Students will be able to

1. Gain knowledge about the fundamentals of Nature's computation techniques
2. Learn a few algorithms inspired by Nature
3. Identify the basic techniques for simulating natural phenomenon
4. Appreciate the efficiency and effectiveness of computation using natural material like DNA, Atomic and Sub-atomic particles

Detailed Contents

UNIT-I

Introduction to Natural Computing: Motivation, three branches of natural computing, when to use natural computing approaches, General concepts and terminology.

Evolutionary Computing: Evolutionary Biology, Principles of genetics, Genetic Algorithm. Pattern recognition example.

UNIT-II

Neurocomputing: Biological nervous system, Artificial Neural Networks- artificial neuron, types of activation function, types of learning (supervised, unsupervised, reinforcement learning) and learning laws, backpropagation.

Swarm intelligence (Social Computing): Ant colonies, Simple Ant Colony Optimization algorithm (S-ACO), Social Adaptation of Knowledge - Particle swarm optimization algorithm

UNIT-III

Immuno Computing: The immune system, An artificial immune system algorithm, From natural to artificial immune systems, Scope of Artificial Immune Systems.

Particle Systems: Principles, basic model of particle systems, pseudo code and examples

UNIT-IV

Fractal Geometry of Nature: Self similarity, fractal dimension, example fractals.

Cellular Automata: Formal definition, one and two dimension cellular automata, Application-generating fractal patterns, scope. L-Systems: Generating words or strings, geometric interpretation, models of plant architecture, scope. Iterated Function Systems(IFS), Fractional Brownian Motion.

UNIT-V

DNA Computing: The DNA molecule, manipulating DNA, Formal models, Universal DNA computers, Classical vs DNA computing, Scope Quantum Computing: Principles of quantum mechanics, qubit, dirac notation, blochsphere notation, quantum gates, quantum parallelism, quantum circuit example – the swap circuit.

Text Books

- Leandro Nunes de Castro, "Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications", Chapman & Hall/CRC, Taylor & Francis Group, 2007

References

1. D. Shiffman, "The Nature of Code – Simulating Natural Systems with Processing", self-published, 2012.
2. A. Brabazon, M. O'Neill, and S. McGarraghy, "Natural Computing Algorithms", Springer, 1st ed., 2015.

Course Code	Course Title					Course Type
CS3235	Information Retrieval					PEC
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- Demonstrate genesis and diversity of information retrieval situations for text and hyper media.
- Describe hands-on experience store, and retrieve information from www using semantic approaches.
- Demonstrate the usage of different data/file structures in building computational search engines.

Course Outcomes

1. Ability to apply IR principles to locate relevant information large collections of data
2. Ability to design different document clustering algorithms
3. Implement retrieval systems for web search tasks.
4. Design an Information Retrieval System for web search tasks.

Detailed Contents

UNIT-I

Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses .**Information Retrieval System Capabilities:**Search Capabilities, Browse Capabilities, Miscellaneous Capabilities.

UNIT-II

Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction

Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

UNIT-II

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages.

Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters.

UNIT-IV

User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext.

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies.

UNIT-V

Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems.

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval.

Text Books

1. Information Storage and Retrieval Systems – Theory and Implementation, Second Edition, Gerald J. Kowalski, Mark T. Maybury, Springer.

References

2. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
3. Information Storage & Retrieval By Robert Korfhage – John Wiley & Sons.
4. Modern Information Retrieval By Yates and Neto Pearson Education.