DEPARTMENT OF INFORMATICS

SCHEME OF INSTRUCTION FOR MCA

SEMESTERS I, II, III & IV

SYLLABI FOR MCA SEMESTERS I AND II

EFFECTIVE FROM 2024-2025

ACADEMIC REGULATIONS 2024 (R24)

Applicable to academic year 2024-2025 & 2025-2026

OMEGA INSTITUTE OF SCIENCE AND TECHNOLOGY

COLLEGE CODE: 2144

Formally Omega PG College- MBA,

(Approved by AICTE & Affiliated to Osmania University)

UGC-AUTONOMOUS, Accredited by NAAC

Master of Computer Applications

MCA

Two years Programme

Scheme of Instruction for MCA Semesters I, II, III & IV
Syllabi for MCA Semesters I and II
Effective from 2024-2025

Academic Regulations 2024 (OM-R24)
Applicable to academic year 2024-2025 & 2025-2026

FACULTY OF INFORMATICS

OMEGA INSTITUTE OF SCIENCE AND TECHNOLOGY

2024-2025

SCHEME OF INSTRUCTION

Master of Computer Applications

SEMESTER-I

Course Code	Course Title	Contact Hours Per Week			Scheme of Evaluation		Credits
		L	T	P	CIE	SEE	
OM-R24PCC101	Mathematical Foundations of Computer Science	3	1	-	30	70	4
OM-R24PCC102	Data Structures using C	3	1	-	30	70	4
OM-R24PCC103	Object Oriented Programming Using Java	3	1	-	30	70	4
OM-R24PCC104	Computer Architecture	3	1	-	30	70	4
OM-R24PCC105	Probability and Statistics	3	1	-	30	70	4
OM-R24MGC106	Managerial Economics and Accountancy	3	1	-	30	70	4
OM-R24LCC151	Data Structures using C Lab	-	-	3	25	**50**	1.5
OM-R24LCC152	Java Programming Lab	-	-	3	25	**50**	1.5
OM-R24HSC151	Soft Skills Lab	-	-	3	25	**50**	1.5
TOTAL		**18**	**6**	**9**	**255**	**570**	**28.5**

Abbreviation	Full Form	Abbreviation	Full Form
PCC	Professional Core Course	**CIE**	Continuous Internal Evaluation
PEC	Professional Elective Course	**SEE**	Semester End Evaluation
MGC	Management Course	**L/T**	Lecture/Tutorial
LCC	Laboratory Core Course	**P**	Practical

SCHEME OF INSTRUCTION

Master of Computer Applications

SEMESTER-II

Course Code	Course Title	Contact Hours Per Week			Scheme of Evaluation		Credits
		L	T	P	CIE	SEE	
OM-R24PCC201	Operating Systems	3	1	-	30	70	4
OM-R24PCC202	Database Management Systems	3	1	-	30	70	4
OM-R24PCC203	Design and Analysis of Algorithms	3	1	-	30	70	4
OM-R24PCC204	Data Engineering with Python	3	1	-	30	70	4
OM-R24PCC205	Machine Learning	3	1	-	30	70	4
OM-R24MGC206	Operations Research	3	1	-	30	70	4
OM-R24LCC251	Operating Systems Lab	-	-	3	25	**50**	1.5
OM-R24LCC252	Data Engineering with Python Lab	-	-	3	25	**50**	1.5
OM-R24LCC253	Database Management Systems Lab	-	-	3	25	**50**	1.5
OM-R24SIP321	Summer Internship	To be completed after the second semester, with credits to be awarded in the third semester.					
TOTAL		**18**	**6**	**9**	**255**	**570**	**28.5**

Abbreviation	Full Form	Abbreviation	Full Form
PCC	Professional Core Course	**CIE**	Continuous Internal Evaluation
PEC	Professional Elective Course	**SEE**	Semester End Evaluation
MGC	Management Course	**L/T**	Lecture/Tutorial
LCC	Laboratory Core Course	**P**	Practical

P.V. Sridhar

Ramay

SCHEME OF INSTRUCTION

Master of Computer Applications

SEMESTER-III								
Course Code		Course Title	Contact Hours Per Week			Scheme of Evaluation		Credits
			L	T	P	CIE	SEE	
	OM-R24PCC301	Software Engineering	3	1	-	30	70	4
	OM-R24PCC302	Computer Networks	3	1	-	30	70	4
	OM-R24PCC303	Artificial Intelligence	3	1	-	30	70	4
	OM-R24PCC304	Web Technologies	3	1	-	30	70	4
Professional Elective- **I**	OM-R24PEC311	Software Quality & Testing	3	1	-	30	70	4
	OM-R24PEC312	Distributed Systems						
	OM-R24PEC313	Internet of Things						
	OM-R24PEC314	Image Processing						
Professional Elective- **II**	OM-R24PEC321	Network Security	3	1	-	30	70	4
	OM-R24PEC322	Cyber Security						
	OM-R24PEC323	Information Retrieval System						
	OM-R24PEC324	Natural Language Processing						
	OM-R24LCC351	Computer Networks Lab	-	-	3	25	50	1.5
	OM-R24LCC352	Software Engineering Lab	-	-	3	25	50	1.5
	OM-R24LCC353	Web Technologies Lab	-	-	3	25	50	1.5
	OM-R24SIP321	Summer Internship	-	-	-	-	50	2
TOTAL			**18**	**6**	**9**	**255**	**620**	**30.5**

Abbreviation	Full Form	Abbreviation	Full Form
PCC	Professional Core Course	**CIE**	Continuous Internal Evaluation
PEC	Professional Elective Course	**SEE**	Semester End Evaluation
MGC	Management Course	**L/T**	Lecture/Tutorial
LCC	Laboratory Core Course	**P**	Practical

SCHEME OF INSTRUCTION

Master of Computer Applications

SEMESTER-IV

Course Code		Course Title	Contact Hours Per Week			Scheme of Evaluation		Credits
			L	T	P	CIE	SEE	
Professional Elective- III	OM-R24PEC411	Block Chain Technologies	3	1	-	30	70	4
	OM-R24PEC412	Big Data Analytics						
	OM-R24PEC413	Cloud Computing						
	OM-R24PEC414	Deep Learning						
Professional Elective- IV	OM-R24PEC421	Distributed Database Systems	3	1	-	30	70	4
	OM-R24PEC422	Digital Forensics						
	OM-R24PEC423	Optimization Techniques						
	OM-R24PEC424	Enterprise Architecture						
Open Elective	OM-R24OEC431	Professional Ethics	3	1	-	30	70	4
	OM-R24OEC432	Constitution of India						
	OM-R24OEC433	Disaster Management						
	OM-R24OEC434	Organization Behaviour						
	OM-R24OEC435	Intellectual Property & Cyber Law						
	OM-R24OEC436	Environmental Science						
OM-R24Proj401		Project Work	-	-	24	50	100	12
TOTAL			9	3	24	140	310	24
CREDITS GRAND TOTAL								111.5

Abbreviation	Full Form	Abbreviation	Full Form
PCC	Professional Core Course	**CIE**	Continuous Internal Evaluation
PEC	Professional Elective Course	**SEE**	Semester End Evaluation
MGC	Management Course	**L/T**	Lecture/Tutorial
LCC	Laboratory Core Course	**P**	Practical

OM-R24PCC101 Mathematical Foundations of Computer Science

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. To learn logic theory and Boolean Algebra related to computer science
2. To understand relations and functions
3. To gain insights into recurrence relation
4. To comprehend algebraic structure
5. To study graph theory and concepts of trees

Course Outcomes

1. Solve logic problems
2. Represent the relations and functions
3. Create recurrence relation
4. Apply algebraic structures
5. Work on various graph and tree concepts

Unit-I

Fundamentals of logic: Basic connective and truth tables, logical equivalence, logical implication, Use of quantifiers, definitions and the proof of theorems. **Set theory:** Sets and subsets, set operations, and the laws of set theory, counting and Venn diagrams. **Properties of the integers:** The well-ordering principle, recursive definitions, division algorithms, fundamental theorem of arithmetic.

Unit-II

Relations and functions: Cartesian product, functions onto functions, special functions, pigeon - hole principle, composition and Inverse functions. **Relations:** Partial orders, equivalence relations and partitions. **Principle of inclusion and exclusion:** Principles of inclusion and exclusion, generalization of principle.

Unit-III

Generating functions: Introductory examples, definitions and examples, partitions of integers. **Recurrence relations:** First-order linear recurrence relation, second-order linear homogeneous recurrence relation with constant coefficients.

Unit-IV

Algebraic structures: Algebraic system general properties, semi groups, monoids, homomorphism, groups, residue arithmetic.

Unit-V

Graph theory: Definitions and examples, subgraphs, complements and graph isomorphism, vertex degree, planar graphs, Hamiltonian parts and cycles. **Trees:** Definition, properties and examples, rooted trees, spanning trees and minimum spanning trees.

Suggested readings

1. Mott Joe L Mott, Abraham Kandel and Theodore P Baker, **Discrete mathematics for computer scientists and mathematicians**, Prentice Hall NJ, Second Edition 2015.
2. Jr. P. Tremblay and R Manohar, **Discrete mathematical structures with applications to computer science**, MCgraw Hill, 1987.
3. R. K. Bisht and H.S Dhami, **Discrete mathematics**, Oxford higher education, 2015.
4. Bhavanari Satyanarayana, Tumurukota Venkata Pradeep Kumar and Shaik Mohiddin Shaw, **Mathematical foundations of computer science**, BSP, 2016.
5. Ralph P. Grimaldi, **Discrete and combinatorial mathematics**, 5th edition, Pearson, 2004.

P. V. Sridha Ramay

OM-R24PCC102 Data Structures using C

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. To learn the features of C
2. To learn the linear and non-linear data structures
3. To explore the applications of linear and non-linear data structures
4. To learn to represent data using graph data structures
5. To learn the basic sorting and searching algorithms

Course Outcomes

1. Implement linear and non-linear data structure operations using C
2. Suggest appropriate linear/non-linear data structures for any given data set.
3. Apply hashing concepts to a given problem
4. Modify or suggest a new data structure for an application
5. Appropriately choose the sorting algorithm for an application.

UNIT-I

Structure of a C program - compilation and linking processes - Constants, Variables - Data Types - Expressions using operators in C - Managing Input and Output operations - Decision Making and Branching - Looping statements. Arrays - Initialization - Declaration - One-dimensional and Two-dimensional arrays. Strings - String operations - String Arrays. Simple programs - sorting - searching - matrix operations.

UNIT-II

Functions - Pass by value - Pass by reference - Recursion - Pointers - Definition - Initialization - Pointer arithmetic. **Structures and unions** - definition - Structure within a structure - Union - Programs using structures and unions - Storage classes, Preprocessor directives.

UNIT-III

Arrays and their representations. **Stacks and Queues** - Applications. **Linked lists** - Single, circular, and doubly linked lists - Applications.

UNIT-IV

Trees - Binary Trees - Binary tree representation and traversals - Applications of trees. Binary Search Trees, AVL trees. **Graph and its representations** - Graph Traversals.

UNIT-V

Linear Search - Binary Search. **Sorting**: Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort. Hashing, Types of Hashing, Collision resolution techniques.

Suggested readings

1. l. Brian W. Kernighan I Dennis Ritchie, **The C Programming language**, Second Edition, Pearson 2015
2. Pradip Dey and Manas Ghosh, **Programming in C**, Second Edition, Oxford University Press, 2011.
3. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, **Fundamentals of Data Structures in C**, Second Edition, University Press, 2008.
4. Mark Allen Weiss, **Data Structures and Algorithm Analysis in C**, Second Edition, Pearson Education, 1996
5. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, **Data Structures and Algorithms**, Pearson Education, I983.

P.V. Sudha

Ramang

OM-R24PCC103 Object Oriented Programming using Java

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. Learn the basics of object-oriented programming
2. Study Java I/O mechanisms
3. Explore Java API
4. Develop graphics-based Java programs
5. Learn the JDBC, servlet, JSP and Swing framework

Course Outcomes

1. Explain OOP features and concepts
2. Write basic Java programs
3. Write I/O programs in Java
4. Use various built-in Java classes and methods
5. Create window-based Java programs

UNIT-I

Object-Oriented System Development: Understanding Object-Oriented Development, understanding object concepts, Benefits of Object-Oriented Development. **Java Programming Fundamentals:** Introduction, Overview of Java, Data Types, Variables and Arrays, Operators, Control Statements, Classes, Methods, Inheritance, Packages and Interfaces, Inner Classes.

UNIT-II

I/O basics, Stream and Byte classes, Character Streams, Reading Console input and output, PrintWriter Class, String Handling, Exception Handling, Multithreaded Programming.

UNIT-III

Exploring Java Language, Collections Overview, Collections Interfaces, Collections Classes, Iterators, Random Access Interface, Maps, Comparators, Arrays, Legacy classes and interfaces, String Tokenizer, BitSet, Date, Calendar, Timer.

UNIT-IV

Introducing AWT Working with Graphics: AWT Classes, Working with Graphics. **Event Handling**: Two Event Handling Mechanisms, The Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces. **AWT Controls**: Control Fundamentals, Labels, Using Buttons, Applying Check Boxes, CheckboxGroup, Choice Controls, Using Lists, Managing Scroll Bars, Using TextField, Using TextArea, Understanding Layout Managers, Menu bars and Menus, Dialog Boxes, FileDialog, Handling events by extending AWT Components, Exploring the controls, Menus, and Layout Manager.

UNIT-V

Introduction to **Swing Package**, Introduction to **JDBC**, JDBC Drivers & Architecture, Connecting to Non-Conventional Databases, Introduction to Servlet, Servlet Life Cycle, Developing and Deploying Servlets, Exploring Deployment, Handling Request and Response, JSP, Introduction to Java Network Programming.

Suggested Readings

1. Herbert Schildt, **The Complete Reference Java**, 9th Edition, Tata McGraw Hill, 2005.
2. Bruce Eckel, **Thinking in Java**, 4th Edition, Pearson Education.
3. Dietel and Dietel, *Java:* **How to Program**, 5th Edition, Prentice Hall.
4. James M. Slack, **Programming and Problem Solving with JAVA**, Thomson Learning, 2002.
5. C. Thomas Wu, **An Introduction to Object-Oriented Programming with Java**, Tata McGraw Hill, 2005.
6. Kathy Sierra, Bert Bates, **Head First Java**, 2nd Edition, A Brain-Friendly Guide, Publisher: O'Reilly Media, February 2005.
7. Jim Keogh, **J2EE**, McGraw Hilll

P.V. Sudha

OM-R24PCC104 Computer Architecture

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. Learn the basics of data representation
2. Study register transfer micro-operations
3. Explore the CPU
4. Comprehend computer arithmetic algorithms
5. Learn I/O organization

Course Outcomes

1. Apply data representation methods
2. Write logic diagrams for micro-operations
3. Write general register organization diagrams
4. Analyze computer arithmetic algorithms
5. Explain I/O organization

UNIT-I

Data Representation: Data types, Complements, Fixed- and Floating-Point representations, and Binary codes. **Overview of Computer Function and Interconnections**: Computer components, Interconnection structures, Bus interconnection, Bus structure, and Data transfer.

UNIT-II

Register Transfer Micro-operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic, Logic and Shift micro-operations, Arithmetic Logic Shift Unit. **Basic Computer Organization and Design**: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory reference instruction, Input-Output and Interrupt.

UNIT-III

Microprogrammed Control: Control memory, Address Sequencing, Micro program example, Design of Control Unit. **Central Processing Unit:** General Register Organization, Stack Organization, Instruction formats, Addressing modes, Data Transfer and Manipulation, and Program control. **Computer Arithmetic:** Addition and Subtraction, Multiplication, Division, and Floating-Point Arithmetic Operations.

UNIT-IV

Memory Organization: Memory Hierarchy, Main Memory, RAM and ROM, Auxiliary memory, Associative memory, Cache memory, Virtual memory, Memory Management hardware.

UNIT-V

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), I/O Processor, Serial Communication. **Pipeline Processing:** Arithmetic, Instruction, and RISC Pipelines. **Assessing and Understanding Performance:** CPU performance and its factors, Evaluating performance.

Suggested Readings

1. Morris Mano, **Computer System Architecture**, Pearson Education India, 3rd Edition, 2007.
2. William Stallings, **Computer Organization and Architecture**, PHI, 7th Edition, 2008.
3. David A. Patterson, John L. Hennessy, **Computer Organization and Design**, Morgan Kaufmann, 5th Edition, 2013.
4. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, **Computer Organization**, Tata McGraw-Hill Education, 5th Edition, 2002.

P.V. Sudha

Ramana

OM-R24PCC105 Probability and Statistics

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. Understand the Linear Algebra concepts through vector spaces.
2. Basic concepts of probability and various discrete and continuous probability distributions.
3. Learn sampling procedures and various kinds of estimation techniques.
4. Learn hypothesis testing and acquire knowledge of basic statistical inference and its applications.
5. Understand the concept of association between two variables and forecast future values by regression equations.

Course Outcomes

1. Understanding of Linear Algebra will boost the ability to understand and apply various data science algorithms.
2. Calculate probabilities by applying probability laws and theoretical results, and gain knowledge of important discrete and continuous distributions and their interrelations with real-time applications.
3. Understand the use of sample statistics to estimate unknown parameters.
4. Become proficient in learning to interpret outcomes.
5. Compute and interpret correlation analysis, regression lines, and multiple regression analysis with applications.

UNIT-I

Vector Spaces - Vector Spaces and Subspaces - Null Spaces, Column Spaces, and Linear Transformations. Linearly Independent Sets - Bases - Coordinate Systems.

UNIT-II

Probability - Basic terminology, three types of probability, probability rules, statistical independence, statistical dependency, Bayes' theorem. **Probability Distributions** - Random variables, expected values, binomial distribution, Poisson distribution, normal distribution, choosing the correct distribution.

UNIT-III

Sampling and Sampling Distributions - Random sampling, Non-Random Sampling distributions, operational considerations in sampling. **Estimation** - Point estimates, interval estimates, confidence intervals, calculating interval estimates of the mean and proportion, t-distribution, determination of sample size in estimation.

UNIT-IV

Testing Hypothesis - One sample tests - Hypothesis testing of the mean when the population standard deviation is known, powers of hypothesis tests, hypothesis testing of proportions, hypothesis testing of means when the standard deviation is not known. **Testing Hypotheses - Two sample tests** - Tests for the difference between means - large sample, small sample, with dependent samples, testing for the difference between proportions - large sample.

UNIT-V

Chi-square and Analysis of Variance - Chi-square as a test of independence, Chi-square as a test of goodness of fit, analysis of variance, inferences about a population variance, inferences about two population variances. **Regression and Correlation** - Simple Regression - Estimation using the regression line, correlation analysis, making inferences about population parameters, limitations, errors, and caveats in regression and correlation analysis. Multiple Regression and correlation analysis. Finding multiple regression equations and making inferences about population parameters.

Suggested Reading

1. David C. Lay, **Linear Algebra and Its Applications**, 4th Edition.
2. Richard I. Levin, David S. Rubin, **Statistics for Management**, Seventh Edition, PHI, 1997.
3. R. D. Sharma, **Theory and Problems of Linear Algebra**, International Publishing House Pvt. Limited, 2011.
4. A. K. Sharma, **Linear Algebra**, Discovery Publishing House Ltd., 2019.
5. Gilbert Strang, **Linear Algebra and Its Applications**, 2010.
6. S. C. Gupta and V. K. Kapoor, **Fundamentals of Mathematical Statistics**, Sultan Chand & Sons, New Delhi.

P.V.Sudha Ramang

OM-R24PCC106 Managerial Economics and Accountancy

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. To learn important concepts of Managerial Economics and apply them to evaluate business decisions.
2. To understand various parameters that determine consumers' behavior.
3. To evaluate the factors that affect production.
4. To understand the concepts of capital budgeting and payback period.
5. To study the concepts of various book keeping methods.

Course Outcomes

1. Apply the fundamental concepts of managerial economics to evaluate business decisions and understand types of demand and factors related to it.
2. Identify different types of markets and determine price-output under perfect competition.
3. Determine working capital requirements and payback.
4. Analyze and interpret financial statements through ratios.
5. Determine double entry book keeping methods

UNIT-I

Meaning and Nature of Managerial Economics: Managerial Economics and its usefulness to engineers, Fundamental Concepts of Managerial Economics - Scarcity, Marginalism, Equi-marginalism, Opportunity costs, Discounting, Time Perspective, Risk and Uncertainty, Profits, Case study method.

UNIT-II

Law of Demand and Supply: Law of Demand, Determinants, Types of Demand; Elasticity of Demand (Price, Income, and Cross-Elasticity); Demand Forecasting, Law of Supply and Concept of Equilibrium.

UNIT-III

Theory of Production and Markets: Production Function, Law of Variable Proportion, ISO quants, Economies of Scale, Cost of Production (Types and their measurement), Concept of Opportunity Cost, Concept of Revenue, Cost-Output relationship, Break-Even Analysis, Price-Output determination under Perfect Competition and Monopoly.

UNIT-IV

Working Capital Management and Capital Budgeting: Concepts, Significance, determination, and estimation of fixed and variable working capital requirements, sources of capital. **Introduction to capital budgeting methods** – traditional and modern methods with problems.

UNIT-V

Accounting: Meaning, Significance, Principles of double-entry book keeping, Journal, Ledger accounts, Subsidiary books, Trial Balance, preparation of Final Accounts with simple adjustments, Analysis and interpretation of Financial Statements through Ratios.

Suggested readings

1. Mehta P.L., **Managerial Economics - Analysis, Problems and Cases**, Sultan Chand & Sons Educational Publishers, 2011.
2. Maheswari S.N., **Introduction to Accountancy**, Vikas Publishing House, 2005.
3. Pandey I.M., **Financial Management**, Vikas Publishing House, 2009.
4. S.P. Jain and K.L. Narang, **Financial Accounting**, Kalyan Publishers, 2018.
5. M. Hanif and A. Mukherjee, **Modern Accountancy**, McGraw Hill, 3rd Edition, 2018.

P.V. Sridhar

Ramany

OM-R24LCC151 Data Structures using C Lab

Credits: 1.5 CIE:25 SEE:50 Hrs/Week: 3P

Course Objectives

1. To understand and implement basic data structures using C.
2. To apply linear and non-linear data structures in problem solving.
3. To learn to implement functions and recursive functions through data structures.
4. To implement searching and sorting algorithms.

Course Outcomes

1. Write basic and advanced programs in C.
2. Implement functions and recursive functions in C.
3. Implement data structures using C.
4. Choose an appropriate sorting algorithm for an application and implement it in a modularized way.

Programs

1. Basic C Programs - looping, data manipulations, arrays.
2. Programs using strings - string function implementation.
3. Programs using structures and pointers.
4. Programs involving dynamic memory allocations.
5. Array implementation of stacks and queues.
6. Linked list implementation of stacks and queues.
7. Applications of Stacks and Queues.
8. Implementation of Trees and Tree Traversals.
9. Implementation of Binary Search Trees.
10. Implementation of Linear search and Binary search.
11. Implementation of Insertion sort, Bubble sort, Quick sort, and Merge sort.
12. Implementation of Hash functions and Collision resolution techniques.

P.V

Ramany

OM-R24LCC152 Java Programming Lab

Course Objectives **Credits: 1.5 CIE:25 SEE:50 Hrs/Week: 3P**

1. Learn how to write simple Java programs.
2. Learn how to write multithreaded programs.
3. Learn how to write GUI programs.
4. Learn how to write serialization programs.
5. Learn how to write programs using the URL class.

Course Outcomes

1. Be able to write simple Java programs.
2. Be able to write multithreaded programs.
3. Be able to write GUI programs.
4. Be able to write serialization programs.
5. Be able to write URL class programs.

Programs

1. Write a program to calculate the salary of n employees using the concept of classes with constructors and methods.
2. Write a program to demonstrate an e-commerce website using inheritance, abstract classes, and dynamic polymorphism.
3. Write a program to demonstrate various arithmetic calculations using packages.
4. Write a program to demonstrate a client-server environment using multithreading.
5. Write a program to demonstrate mutual exclusion using thread synchronization.
6. Write a program to demonstrate a Linked List class.
7. Write a program to demonstrate HashSet and Iterator classes.
8. Write a program to demonstrate Enumeration and Comparator interfaces.
9. Write a program to accept data and display output in key-value pairs.
10. Write a program to create a registration form with different controls, menus, and demonstrate event handling.

P.V. Sridha Ramaney

11. Write a program to copy data from one file to another file.
12. Write a program to merge the contents of two files and display output on the console.
13. Write a program to illustrate serialization.
14. Write a program to retrieve a web page using the URL class.
15. Write a program to load and display image and perform gray scale.

P. V. Sudha Ramanay

OM-R24HSC153 Soft Skills Lab

Credits: 1.5 CIE:25 SEE:50 Hrs/Week: 3P

Course Objectives

1. Learn conversational skills.
2. Learn reading strategies.
3. Learn time management.
4. Learn stress management.
5. Learn career planning.

Course Outcomes

1. Express conversational skills.
2. Specify reading strategies.
3. Perform time management.
4. Perform stress management.
5. Explore career planning.

Activities

1. Conversation skills, listening dialogues from TV/radio/Ted Talks/Podcasts.
2. Group discussion.
3. Interview Skills, Making Presentation.
4. listening to lectures and News programmes, listening to talk shows.
5. Watching videos on interesting events on Youtube,
6. Reading different genres of tests ranging from newspapers to philosophical treatises.
7. Reading strategies – graphic organizers, Reading strategies – summarizing
8. Reading strategies – interpretation reports.
9. Writing cover letters, resumes.
10. Writing for publications, letters, memos, emails, and blogs.
11. Civil service (language-related), verbal ability.
12. Motivation, self-image.

13. Goal setting, managing changes.
14. Time management, stress management.
15. Leadership traits.
16. Teamwork.
17. Career and life planning.
18. Multiple intelligences.
19. Emotional intelligence.
20. Spiritual quotient (ethics).
21. Intercultural communication.
22. Creative and critical thinking.
23. Learning styles and strategies.

Suggested readings

1. **Business English Certificate materials**, Cambridge University Press.
2. **Graded Examinations in Spoken English and Spoken English for Work downloadable materials from Trinity College**, London.
3. **International English Language Testing System Practice Tests**, Cambridge University Press.
4. **Interactive Multimedia Programs on Managing Time and Stress.**
5. **Personality Development (CD-ROM)**, Times Multimedia, Mumbai.
6. Robert M. Sherfield, **Developing Soft Skills**, 4th Edition, Pearson Education, 2009.

Web Resources:

http://www.slideshare.net/rohitjsh/presentation-on-group-discussion

http://www.washington.edu/doit/TeamN/present_tips.html

http://www.oxforddictionaries.com/words/writing-job-applications

http://www.kent.ac.uk/careers/cv/coveringletters.htm

http://www.mindtools.com/pages/article/newCDV_34.htm

P.V. Sridhar Ramany

OM-R24PCC201 Operating Systems

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. To gain an understanding of operating systems, with a focus on Unix operating system.
2. To comprehend the details of process.
3. To learn about the types and architecture of computer memory.
4. To study file systems and their implementation.
5. To apply operating system concepts through case studies.

Course Outcomes

1. Explain operating systems and Unix OS, and illustrate the workings of various OS components.
2. Analyze the process, its states, and process scheduling algorithms.
3. Demonstrate paging, demand paging, page replacement, and segmentation with illustrations.
4. Elaborate on file access, allocation methods, and mass storage structures.
5. Describe the concrete implementations of the Linux system and Windows 7.

UNIT-I

Unix: Introduction, commands, file system, security and file permissions, regular expressions and grep, shell programming, awk. **Introduction to Operating Systems:** OS structure and strategies, process concepts, multithreaded programming, process scheduling, process synchronization, deadlocks.

UNIT-II

Memory Management Strategies with Example Architectures: Swapping, contiguous allocation, paging, segmentation, segmentation with paging. **Virtual Memory Management:** Demand paging, page replacement, thrashing.

P.V.Sridhar Ramang

UNIT-III

File System Interface: File concepts, access methods, and protection. **File System Implementation:** File system structure, allocation methods, directory implementation, mass storage structures, I/O systems.

UNIT-IV

System Protection: Principles and domains, access matrix and implementation, access control and access rights, capability-based systems, language-based protection. **System Security:** Problem, program threats, cryptography, user authentication, implementing security defenses, firewalls, computer security classification.

UNIT-V

Case Studies: The Linux System-Design principles, Kemel modules, Process management, Scheduling, Memory management, File systems, Input and Output, Inter process. communication. **Windows 7** - Design principles, System components, Terminal services and fast user switching File systems, Networking, Programmer interface.

Suggested readings

1. Abraham Silberschatz, Peter B. Galvin, **Operating System Concepts**, 9th edition, Wiley, 2016.
2. William Stallings, **Operating Systems: Internals and Design Principles**, 8th edition, Pearson, 2014.
3. Andrew S. Tanenbaum, **Modern Operating Systems**, 4th edition, Pearson, 2016.

OM-R24PCC202 Database Management Systems

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. Introduce database concepts along with ER modeling.
2. Learn about relational databases and SQL query language.
3. Define advanced SQL.
4. Study database transactions and explore concurrency concepts.
5. Introduce NoSQL.

Course Outcomes

1. Explain database concepts and model requirements using the ER model.
2. Formulate relational algebra queries from textual specifications.
3. Write SQL queries for given problems.
4. Elaborate on indexing and hashing, and describe concurrency control concepts.
5. Understand NoSQL technology.

UNIT-I

Introduction: Database system applications, purpose of database systems, views of values, nested sub-queries, complex queries, views, modification of the database, joined relations data, database languages, relational databases, database design, object-based and semi-structured databases, data storage and querying, transaction management, data mining and analysis, database architecture, database users and administrators. **Database Design and the E-R Model:** Overview of the design process, the entity-relationship model, constraints, entity-relationship diagrams, entity-relationship design issues, weak entity sets, extended E-R features, database design for a banking enterprise, reduction to relational schemas, and other aspects of database design.

UNIT-II

Relational Model: Structure of relational databases, fundamental relational algebra operations, additional relational algebra operations, extended relational algebra operations, null values, modification of databases. **Structured Query Language (SQL):** Data definition, basic structure of SQL queries, set operations, aggregate functions, null.

UNIT-III

Advanced SQL: SQL data types and schemas, integrity constraints, authorization, embedded SQL, dynamic SQL, functions and procedural constructs, recursive queries, and advanced SQL features. **Relational Database Design:** Features of good relational design, atomic domains and first normal form, functional dependency theory, decomposition using functional dependencies.

UNIT-IV

Indexing and Hashing: Basic concepts, ordered indices, B+ tree index files, B tree index files, multiple-key access, static hashing, dynamic hashing, comparison of ordered indexing and hashing, bitmap indices. **Index Definition in SQL Transactions**: transaction concepts, transaction states, implementation of atomicity and durability, concurrent executions, serializability, recoverability, implementation of isolation, testing for serializability.

UNIT-V

Concurrency Control: Lock-based protocols, timestamp-based protocols, validation-based protocols, multiple granularity, multi-version schemes, deadlock handling, insert and delete operations, weak levels of consistency, concurrency of index structures. **Recovery System:** Failure classification, storage structure, recovery and atomicity, log-based recovery, recovery with concurrent transactions, buffer management, failure with loss of nonvolatile storage, advanced recovery techniques, remote backup systems. **NoSQL:** Need for NoSQL, aggregate data models, more details on data models, distribution models, consistency, version stamps, map-reduce, key-value databases, document databases, column-family stores, graph databases, schema migrations.

Suggested readings

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, **Database System Concepts**, McGraw-Hill International Edition, 6th Edition, 2010.
2. Ramakrishnan, Gehrke, **Database Management Systems**, McGraw-Hill International Edition, 3rd Edition, 2003.
3. Elmasri, Navathe, Somayajulu, **Fundamentals of Database Systems**, Pearson Education, 4th Edition, 2004.
4. Shashank Tiwari, **Professional NoSQL**, 1st Edition, Wiley Publishers, 2011.

P.V. Sridhar Ramany

OM-R24PCC203 Design and Analysis of Algorithms

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. Learn algorithms time complexity
2. Learn divide and conquer approach
3. Learn greedy method
4. Learn dynamic programming
5. Learn backtracking

Course Outcomes

1. Carry out algorithms time complexity
2. Explain divide and conquer approach
3. Illustrate greedy method
4. Elaborate dynamic programming
5. Explore backtracking

Unit-I

Introduction to Algorithms: Algorithm Specifications, Performance Analysis, Randomized Algorithms. **Elementary Data Structures:** Stacks and Queues, Trees, Dictionaries, Priority Queues, Sets and Disjoint Set Union, Graphs.

Unit-II

Divide and Conquer: Binary Search, Finding the Maximum and Minimum, Merge Sort; Quick Sort, Selection sort, Strassen's Matrix Multiplication, Convex Hull. **The Greedy Method:** Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees, Singles Source Shortest Paths.

Unit-III

Dynamic Programming: General Method, Multistage Graphs, All-Pairs Shortest Paths, Single-Source Shortest Paths, Optimal Binary Search Trees, 0/1 Knapsack, The Traveling Salesperson Problem. **Basic Traversal and Search Techniques:** Techniques for Binary Trees, Techniques for Graphs, Connected Components and Spanning Trees, Biconnected Components and DFS.

Unit-ƅ

Back Tracking: General Method, 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack Problem. **Branch-Bound:** The Method, 0/1 Knapsack Problem, Traveling Sales Person.

Unit-V

NP-Hard and NP-Completed Problems: Basic Concepts, Cook's Theorem, NP-Hard Graph Problems, NP-Hard Scheduling Problems, NP-Hard Code Generation, Some Simplified NP-Hard Problems.

Suggested Readings

1. E Horowitz, S Sahni, S Rajasekaran, **Fundamentals of Computer Algorithms**, Second Edition, Universities Press 2007.
2. R.Pannerselvan, **Design Analysis and Algorithm**, PHI, 2007.
3. Hari Mohan Pandey, **Design analysis and Algorithm**, University Science Press, 2009.
4. TH Cormen, CE Leiserson, RL Rivert, C Stein, **Introduction to Algorithms**, Third Edition, PHI, 2010.

P.V.

Ramaraj

OM-R24PCC 204 Data Engineering with Python

Course Objectives **Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T**

1. How to extract raw data
2. Clean the data
3. Perform transformations on data
4. Load data and visualize the data

Course Outcomes

1. Understand the basics of Python Programming Language
2. Handle different types of files and work with text data
3. Use regular expression operations
4. Use relational databases via SQL
5. Use tabular numeric data
6. Use the data structures: data series and frames
7. Use PyPlot for visualization
8. Use Python for basic Machine Learning

Unit-I

Introduction, parts of Python Programming Language, Control Flow Statements, Functions, Strings [Reference 2 - Chapter 1 or Chapter 5]

Unit-II

Lists, Dictonaries, Tuples and sets, Files, Regular Expressions [Reference 2- Chapter 6 to Chapter 10]

Uuit-III

Introduction to Data Science [Reference 2-Chapter 12], **Data Science:** Data Analysis Sequence, Data Acquisition Pipeline, Report Structure [Reference 1(Chapter 1-Unit 3)]].

Files and Working with Text Data: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules. [Reference 2, Chapter 9)] **Working with Text Data:** JSON and XML in Python [Reference 2, Section12.2]. **Working with Text Data:** Processing HTML Files, Processing Texts in Natural Languages [Reference 1 (Chapter3 -Unit 13, and Unitl6). **Regular Expression Operations:** Using Special Characters, Regular Expression Methods, Named Groups in Python Regular Expressions, Regular Expression with glob Module [Reference 2-Chapter l0]

Unit - IV

Working with Databases: Setting Up a MySQL Database, using a MySQL Database: Command Line, Using a MySQL Database, Taming Document Stores: MongoDB [Reference 1 (Chapter4-Unitl7toUnit20)]. **Working with Data Series and Frames:** Pandas Data Structures, Reshaping Data, Handling Missing Data, Combining Data, Ordering and Describing Data, Transforming Data, Taming Pandas File I/O [Reference 1 (Chapter 6-Unit 31 to Unit 37)]. **Plotting:** Basic Plotting with PyPlot, Getting to Know Other Plot Types, Mastering Embellishments, Plotting with Pandas [Reference 1 (Chapter8-Unit 41 to Unit 44)]

Unit - V

Probability and Statistics: Reviewing Probability Distributions, Recollecting Statistical measures, Doing Stats the Python way [Reference I (Chapter9-Unit 45 to Unit 47)]. **Machine Learning**: Designing a Predictive Experiment, fitting a linear regression, Grouping Data with K-means Clustering. Surviving in Random Decision Forests. [Reference 1(Chapter 10 – Unit 48 to Unit-5l)]

Suggested Readings

1. Data Science Essentials in Python: Collect, Organize, Explore, Predict, Value. Dmitry Zinoriev, The Pragmatic Programmers LLC,2016
2. Introduction to Python Programming. Gowrishankar S., Veena A. CRC Press, Taylor & Francis Group, 2019

P.V.

3. Python for Everybody: Exploring Data Using Python 3. Charles R Severance,2016
4. Python Data Analytics - Data Analysis and Science using Pandas, matplotlib and the Python Programming Language. Fabio Nelli, Apress, 2015
5. Website Scraping with Python. Using Beautiful-soup and Scrapy. GaborLaszloHajba, Apress, 2018
6. Machine Learning with Python Cookbook:.Practical Solutions from Preprocessing to Deep Learning. Chris Albon, O'Reilly 2018.

P.V. Sridhar Ramany

OM-R24PCC205 Machine Learning

Credits: 4 CIE:30 SEE:70 Hrs/Week: 3L+1T

Course Objectives

1. Learn regression techniques
2. Learn dimensionality reduction methods
3. Learn classification schemes
4. Learn clustering mechanisms
5. Learn evaluation metrics

Course Outcomes

1. Solve regression problems
2. Apply dimensionality reduction methods
3. Analyze classification schemes
4. Explore clustering mechanisms
5. Explain evaluation metrics

Unit-I

Basic Maths: Probability, Linear Algebra, Convex Optimization **Background:** Statistical Decision Theory, Bayesian Learning (ML, MAP, Bayes estimates, Conjugate priors)

Unit-II

Regression: Linear Regression, Ridge Regression, Lasso, **Dimensionality Reduction:** Principal Component Analysis, Partial Least Squares

Unit-III

Classification: Linear Classification, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Perceptron, Support Vector Machines, Kernels, Artificial Neural Networks, Back Propagation, Decision Trees, Bayes Optimal Classifier, Naive Bayes.

Unit-IV

Evaluation measures: Hypothesis testing, Ensemble Methods, Bagging, Adaboost Gradient Boosting, Clustering, K-means, K-medoids, Density-based Hierarchical, Spectral.

Unit-V

Introduction to reinforcement learning, the learning task, Q learning and illustrative example. **Graphical models**: Bayesian networks, use cases of various ML algorithms in manufacturing, retail, transport, healthcare, weather, insurance

Suggested Readings

1. Ethem Alpaydin, **Introduction to Machine Learning**, 3rd edition, Adaptive Computation and Machine Learning Series, The MIT Press, 2004.
2. Tom M. Mitchell, **Machine Learning**, McGraw Hill Education, 2013

P.V. Sudha

Ramanay

OM-R24LCC251 Operating Systems Lab

Credits: 1.5 CIE:25 SEE:50 Hrs/Week: 3P

Course Objectives

1. Learn shell commands and scripting
2. Learn CPU scheduling algorithms
3. Learn memory management algorithms
4. Learn synchronization problems
5. Explore file allocation strategies and disk scheduling algorithms

Course Outcomes

1. Be able to execute shell commands and write shell scripts
2. Be able to write programs on CPU scheduling
3. Be able to create memory management algorithms
4. Be able to execute programs to demonstrate synchronization problems
5. Be able to implement file allocation methods and be able to create disk scheduling algorithms

Programs

1. Unix Shell Commands
   a) File handling commands
   b) Directory handling commands
   c) General purpose commands
2. Unix Shell Scripts
   a) Print Multiplication table of a given number using all loops
   b) Perform all arithmetic operations
   c) Print the type of a file
   d) Rename all files whose names end with .c as .old
   e) Display the number of lines in each of text file in a given dir

3. Simulate the following CPU scheduling algorithms.
    a. FCFS
    b. SJF
    c. Round Robin
    d. Priority.
4. Write a C program to simulate producer-consumer problem using Semaphores.
5. Write a C program to simulate the concept of Dining-philosophers problem.
6. Simulate MVT and MFT.
7. Write a C program to simulate the following contiguous memory allocation techniques
    a. Worst fit
    b. Best fit
    c. First fit.
8. Simulate following page replacement algorithms
    a. FIFO
    b. LRU
    c. OPTIMAL
9. Simulate following File Organization Techniques
    a. Single level directory
    b. Two level directory
10. Simulate following file allocation strategies
    a. Sequential
    b. Indexed
    c. Linked.
11. Simulate Bankers Algorithm for Deadlock Avoidance.
12. Simulate Bankers Algorithm for Deadlock Prevention.
13. Write a C program to simulate disk scheduling algorithms.
    a. FCFS
    b. SCAN
    c. C-SCAN

P. V. Sridhar

Ramany

OM-R24LCC252 Data Engineering with Python Lab

Credits: 1.5 CIE:25 SEE:50 Hrs/Week: 3P

Course objectives

1. Understand the process of Importing and Exporting the data.
2. Learn how to collect, store and manage data from multiple data sources.
3. Know the insights of data using statistical methods
4. Identify different techniques for data analysis and data visualization.
5. Put into practice the ETL (extract, transform, load) pipeline which will extract raw data, clean the data, perform transformations on data, load data and visualize the data.

Course Outcomes

1. Demonstrate various data types in python and develop programs using files, exception handling, functions, classes in Python.
2. Examine the process for importing and exporting the data.
3. Apply appropriate data collection and pre-processing methods.
4. Identify different data analysis Techniques suitable for a given application
5. Demonstrate data visualization techniques for Data Analysis.

Libraries

In this course students are expected to extract, transform and load input data that can be text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc. For doing this, they should learn the following Python libraries/modules: pandas, numpy, BeautifulSoup, pymysql, pymongo, nltk, matplotlib

Datasets

For this laboratory, appropriate publicly available data-sets, can be studied and used. Example:

MNIST (http://yann.lecun.com/exdb/mnist/),

UCI Machine Learning Repository(https://archive.ics.uci.edu/ml/datasets.html),

Kaggle (https://www.kaggle.com/datasets)

Twitter Data

Exercises

1. Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.
2. Write programs for reading and writing binary files
3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions.
4. Design a relational database for a small application and populate the database, Using SQL do the CRUD (create, read, update and delete) operations
5. Create a python MongoDB client using the Python module pymango. Using a collection object Practice functions for insertion, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes
6. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements.
7. Write programs to use the pandas data structures: Frames and series as Storage containers and for a variety of data-wrangling operations, such as:
    - Single-level and hierarchical indexing
    - Handling missing data
    - Arithmetic and Boolean operations on entire columns and tables
    - Database-type operations (such as merging and aggregation)
    - Plotting individual columns and whole tables
    - Reading data from files and writing data to files

Additional Exercises (for learning and practice)

1. Introduction to Python Programming
   a) Running instructions in Interactive interpreter and a Python Script.
   b) Write a program to purposefully raise Indentation Error and Correct it
   c) Write a program to compute distance between two points taking input from the user
   d) Write a program add python that takes 2 numbers as command line arguments and prints its sum.
   e) Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects
   f) Write a Program for checking whether the given number is an even number or not.
2. Control Structures, Lists
   a) Program to find the largest three integers using if-else
   b) Program that receives a series of positive numbers and display the numbers in order and their sum
   c) Program to find the product of two matrices and
   d) Program to display two random numbers that are to be added, the program should allow the student to enter the answer. If the answer is correct a message of congratulations should be displayed. If the answer is incorrect the correct answer should be displayed.
   e) Using a for loop, write a program that prints out the decimal equivalents of 1/2,1/3,1/4, .1/10.
   f) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
3. Functions and Recursion - Write recursive and non-recursive functions for the following
   a) To find GCD of two integers
   b) To find the factorial of positive integer
   c) To print Fibonacci Sequence up to given number n
   d) To display prime number from2 to n.
   e) Function that accepts two arguments: a list and a number n. It displays all of the numbers in the list that are greater than n

f) Functions that accept a string as an argument and return the number of vowels and consonants that the string contains

4. Files, Exceptions, Lists, Sets, Random Numbers
   a) Program to write a series of random numbers in a file from 1 to n and display.
   b) Program to write the content in a file and display it with a line number followed by a colon.
   c) Program to display a list of all unique words in a text file.
   d) Program to analyze the two text files using set operations.
   e) Write a program to print each line of a file in reverse order.
   f) Write a program to count frequency of characters in a given file. Can you use character frequency total whether the given file is a Python program file, C program file or a text file?
   g) Write a program combine lists that combines these lists in to a dictionary.
5. Object Oriented Programming
   a) Program to implement the inheritance.
   b) Program to implement the polymorphism.
6. Demonstrate data analysis using NumPy
   a) Create an array of 10 zeros
   b) Create an array of even integers upto 50
   c) Create a 3x3 matrix
   d) Generate an array of 25 random numbers sampled from a standard normal distribution.
   e) Create an array of 20 linearly spaced points between 0 and I
   f) Demonstrate slicing and indexing operations
   g) Get the sum of all columns in matrix
7. Write a Program in Python to create and combine student and subject data frames in Pandas.
8. Create a data frame 'Book' that contains three vectors [Name, Price, Author]. Convert this data frame into a matrix and list the object using the operator 'as'.

9. Performing Exploratory data analysis on web scraped data of 2021-22 NBA player stats (http://www.basketball-reference.com/)
   a) Perform data cleaning
   b) Handle missing values by replacing with 0
   c) Write to CSV file
   d) Which player scored the most points per game?
   e) Which player had the highest 3-point field goals per game?
   f) Demonstrate Group By() function
10. Data visualization through Seaborn for the above program 9.
    a) Box plot of points scored grouped by position
    b) Compute the correlation matrix
11. To determine the mean of a set of numbers. To plot the numbers in a bar plot and have a straight line run through the plot at the mean.
12. To determine the median of a set of numbers. To plot the numbers in a barplot and have a straight line run through the plot at the median.
13. To determine the standard deviation. To plot the numbers in a bar plot and have a straight line run through the plot at the mean and another straight line run through the plot at mean standard deviation.

More dataset to perform data analysis

Source of the Data: https://www.kaggle.com/chirin/africa-economic-banking-and-systemic-crisis-data/downloads/africa-economic-banking-and-systemic-crisis-data.zip/1

Data set: https://www.kaggle.com/khalidative/crimeanalysis

P.V.

Ramaraj

OM-R24LCC253 Database Management Systems Lab

Credits: 1.5 CIE:25 SEE:50 Hrs/Week: 3P

Course Objectives

1. Learn SQL queries
2. Learn PL/SQL stored procedures
3. Learn Triggers
4. Learn report generation methods
5. Learn database application creation

Course Outcomes

1. Write SQL queries
2. Write stored procedures
3. Write triggers
4. Use file locking and table locking facilities
5. Create small full-fledged database application

Creation of database (exercising the commands for creation)

1. Simple to Complex condition query creation using SQL plus.
2. Usage of Triggers and Stored Procedures.
3. Creation of Forms for Student information, Library information, Pay roll etc.
4. Writing PL/SQL procedures for data validation.
5. Report generation using SQL reports.
6. Creating password and security features for applications.
7. Usage of File locking, Table locking facilities in applications.
8. Creation of small full- fledged database application spreading over 3 sessions.

Note: The creation of sample database for the purpose of the experiments is expected to be pre-decided by the instructor.

Ramang

P.V. Sudha

OM-R24SIP321 Summer Internship/Mini Project

Program Description

The Internship Program/ Mini Project allows MCA students to gain practical experience in the workplace before receiving their graduate degrees. The internship is a required academic course. The student identifies companies willing to hire him/her on a full-time basis for 6-week period (minimum required), usually in the summer. The Internship Program supervises the students and awards academic credits (2) in third semester. Those students who wish to do a Mini Project can use Problem statements and Data Sources from good quality sources and implement a solution. The student will be evaluated based on the working system that is presented in Semester III of this programme.

Intended Learning Outcomes

1. Communicate a practical understanding of how a technology actually operates.
2. Demonstrate the ability to integrate and apply theoretical knowledge and skills developed in various courses to real-world situations in a business organization.
3. Exhibit the ability to effectively work in a professional environment and demonstrate work ethic and commitment in a work-based environment.
4. Demonstrate the ability to successfully complete internship assignments.
5. Reflect on personal and professional development needs and set strategic goals for advancing along an intended career path.
6. Communicate effectively in a professional environment in both English and regional language, orally and in writing.

P.V. Sridha

Ramana