Jyoti Nivas College Autonomous

Post Graduate Centre
Hosur Road, Bangalore -95

Phone No: 255012143 Fax: 25533886
e-mail:jncpgcourses@gmail.com
Website: www.jyotinivas.org

Proposed Scheme & Syllabus

M. Sc in Computer Science

w. e. f. 2023-25 Batch onwards

CONTENTS

M.Sc programme in Computer Science is designed to provide an insight into computing through advanced concepts, principles, strategies and skills supplemented with practical knowledge to effectively develop and work with a range of technologies to build systems and applications that help apply in real-time computing environments. The programme combines strong fundamentals, projects, and team-oriented activities, leading to a holistic education.

PROGRAMME OBJECTIVES:

- Provide a strong foundation in core areas of Computer Science
- Impart knowledge of latest developments in the field of IT
- Identify research gaps and provide innovative solutions
- Adhere to social, ethical, and cyber values

PROGRAMME SPECIFIC OUTCOMES:

In the span of four semesters, the students are well equipped with programming skills, analytical ability, research skills and strong knowledge in the domain. The students will be able to:

- Analytical and critical thinking abilities to solve real time problems
- Design, develop and maintain software applications for different verticals
- Ability to communicate ideas and solutions and work in a collaborative manner
- Amenable to life-long learning of new tools and technology

PROGRAMME STRUCTURE

- The M.Sc programme has a curriculum with syllabi consisting of (i) Core courses, (Theory & Practical) (ii) Skill Enhancement Courses, (iii) Discipline specific Elective Courses (iv) Open Electives and (v) Research Project work.
- In addition, add-on courses, group activities, soft skills, communication skills and value added training are provided based on the need.
- The total number of credits assigned to the course is 98 and the credits per paper are distributed as follows:
    - 4 credits per core theory courses with 4 hours of lecture/tutorial per week
    - 2 credits per practical courses with 4 hours of practical/project per week
    - 4 credits for discipline specific elective courses with 4 hours of lecture/tutorial per week
    - 4 credits per open elective paper with 4 lecture hours per week
    - 12 credits for Research Project / Internship

- Core courses are compulsory subjects offered by the department. Total credits for Core Courses including theory, practical papers, skill enhancement courses and projects are 78.
- Elective courses may be chosen by the student from the list of Electives offered by the department. Total credits for Elective Courses are 16.
- Open Elective has to be taken from other post graduate disciplines. Credit for open Elective course is 4.

REGULATION AND SCHEME

1. **Eligibility:**
   a. B.Sc. (Computer Science) or BCA with Mathematics as one of the subject and at least 50% aggregate marks of all optional subjects (throughout 3 years B.Sc. / BCA course), B.Sc. PCM with PG Diploma / Certificate in Computer Science of duration one year from a recognized University.
   b. The minimum requirement for SC / ST candidates are relaxed in accordance with University regulations.

2. **Duration of the Course: 2** years (**4** semesters)

3. **Medium of Instruction:** The medium of Instruction and Examination shall be in English.

4. **Proposed Intake:** 30

5. **Mode of Admission:** A category wise merit list will be prepared with marks obtained in all optional in all the three years

6. **Evaluation Procedure:**

   a. Continuous Internal Assessment for theory (CIA): 30 Marks

Component	Marks
Mid Term Examination	15
Assignment / Presentation / Case Study / Mini Project	10
Group Activity	5
Total	**30**

   b. End Semester Examination for theory (ESE) : 70 Marks

   c. Continuous Internal Assessment for Practical (CIA): 15 Marks

Component	Marks
Mid Term Examination	10
Review of the work done & Documentation	5
Total	**30**

   d. End Semester Examination for Practical : 35 Marks

   e. Students should secure a paper minimum of 40% each in CIA, end semester theory / practical examination and an aggregate of 50% in a semester.

GRADUATE ATTRIBUTES

1. Domain proficiency

Generally defined, domain proficiency implies knowledge and understanding of the essential aspects of a specific field of inquiry. Domain proficiency is used to evaluate the inputs, guide the process, and evaluate the end products within the context of value and validity.

2. Applied knowledge

It's about the description and demonstration of procedures and tasks that are learnt in the classroom. The knowledge could be applied to develop or revise procedures or algorithms to address a problem / situation.

3. Joint Effort

Joint effort is the collaborative effort of a group to achieve a common goal or to complete a task in the most effective and efficient way. This concept is seen within the greater framework of a team, which is a group of interdependent individuals who work together towards a common goal.

4. Problem solving

Problem solving is the act of defining a problem; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution.

5. Design and development

The design and development process should examine all the potential challenges and hurdles the students will need to overcome to develop an effective application. It involves defining the specification and the design that is simple yet creative and user friendly.

6. Research skills

Research skills refer to the ability to search for, locate, extract, organise, evaluate and use or present information that is relevant to a particular topic. It involves intensive search, investigation, and critical analysis, usually in response to a specific research question or hypothesis.

7. Application Governance.

Application Governance refers to the branch of project management dedicated to the planning, scheduling, resource allocation, execution, tracking and delivery of software and web projects.

8. Usage of modern tools.

Introducing the latest and open source tools that are available to handle the application development.

9. Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

SEMESTER-WISE SCHEDULE

Semester	Course Code	Course Title	Hours / Week (32)			Credits	Marks		
			L	T	P		CIA	ESE	Total
I	23MCS101	Advanced Operating Systems	4	–	–	4	30	70	100
	23MCS102	Theory of Computation	4	–	–	4	30	70	100
	23MCS103	Advanced Computer Architecture	4	–	–	4	30	70	100
	23MCS104	Data Structures	4	–	4	6	30+15	70+35	150
	23MCS105	Object Oriented Programming using Java	4	–	4	6	30+15	70+35	150
	23MCS106	Python Programming	–	–	4	2	15	35	50
		TOTAL	**20**	**0**	**12**	**26**	**195**	**455**	**650**
II	23MCS201	Artificial Intelligence	4	–	–	4	30	70	100
	23MCS202	Advanced Computer Networks	4	–	–	4	30	70	100
	23MCS203	Elective - I	4	–	–	4	30	70	100
	23MCS204	Advanced Database Management System	4	–	4	6	30+15	70+35	150
	23MCS205	Web Technologies	4	–	4	6	30+15	70+35	150
	23MCS206	Agile Software Engineering	–	–	4	2	15	35	50
		TOTAL	**20**	**0**	**12**	**26**	**195**	**455**	**650**
III	23MCS301	Elective - II	4	–	–	4	30	70	100
	23MCS302	Elective - III	4	–	–	4	30	70	100
	23MCS303	Open Elective	4	–	–	4	30	70	100
	23MCS304	Advanced Algorithms	4	–	4	6	30+15	70+35	150
	23MCS305	Data Analytics	4	–	4	6	30+15	70+35	150
	23MCS306	Specialization Project	–	–	4	2	15	35	50
		TOTAL	**20**	**0**	**12**	**26**	**195**	**455**	**650**
IV	23MCS401	Cloud Computing	4	–	–	4	30	70	100
	23MCS402	Research Methodology	4	–	–	4	30	70	100
	23MCS404	Research Project / Internship	–	–	–	12	100	200	300
		TOTAL	**8**	**0**	**0**	**20**	**160**	**340**	**500**

Total Credits (1- 4 semesters): 98

List of Discipline Specific Electives

Electives - AI & ML	
23MCS203A	Digital Image Processing
23MCS301A	Machine Learning Techniques
23MCS302A	Natural Language Processing
23MCS306A	Specialization Project
Electives - Cyber Security	
23MCS203B	Web Application Security
23MCS301B	Ethical Hacking
23MCS302B	BlockChain Technology
23MCS306B	Specialization Project
Electives - Data Science	
23MCS203C	Statistical Concepts for Data Science
23MCS301C	Machine Learning Techniques
23MCS302C	Data Visualization Techniques
23MCS306C	Specialization Project

Elective I	
23MCS203A	Digital Image Processing
23MCS203B	Web Application Security
23MCS203C	Statistical Concepts for Data Science
Elective II	
23MCS301A	Machine Learning Techniques
23MCS301B	Ethical Hacking
23MCS301C	Machine Learning Techniques
Elective III	
23MCS302A	Natural Language Processing
23MCS302B	BlockChain Technology
23MCS302C	Data Visualization Techniques

List of Open Elective Courses available for M,Sc (CS) Students

- Personal Wealth Management
- Approaches to Texts
- Life Style Management
- Chemistry in Daily Life

Open Elective offered to other Discipline Students

- Cyber Security

Add-on Certificate Courses

- Logic N Life
- Soft Skills & Communication Skills
- Academic Writing
- Recent Technologies

Semester I

23MCS101 ADVANCED OPERATING SYSTEMS

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: This course provides the basics of Operating System, key concepts from Distributed Operating System, real time and mobile operating systems. The students will be equipped with the knowledge about the usage of Operating Systems at different levels.

Learning Outcomes

After completion of this course, the student will be able to:

- Discuss the various synchronization, scheduling and memory management issues
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Identify the different features of real time and mobile operating systems
- Gain knowledge on Operating Systems used in embedded systems.

UNIT I: Introduction: Types of Operating Systems, Operating System Structures, Components & Services, Process Management: Process Concept, Process Scheduling, Co – Operating process, Threads, Inter process communication. Process Synchronization and deadlocks: The Critical Section Problem, Synchronization hardware, Semaphores, Classical problems of synchronization. Dead locks – system model, Characterization, Dead lock prevention, avoidance and detection, Recovery from dead lock. **(12)**

UNIT II: Memory management: Background – Swapping - Contiguous memory allocation – Paging – Segmentation - Segmentation with paging. Virtual memory: Background - Demand paging - Process creation - Page replacement -Allocation of frames - Thrashing. Storage Management: directory and disk structure, file system mounting, file sharing, protection.

Mass storage structure: overview, disk structure, disk attachment, disk scheduling, disk management, swap space management, RAID structure. **(12)**

UNIT III: Distributed Operating System: Issues in Distributed Operating System – Architecture – Communication Primitives – Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, and Election Algorithms. Lamport's Logical clocks – Agreement Protocols. **(12)**

UNIT IV: Distributed Shared Memory: Introduction, General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, and Advantages of DSM. **(12)**

UNIT V: Mobile and Real Time Operating Systems: Basic Model of Real Time Systems – Characteristics – Applications of Real Time Systems – Real Time Task Scheduling – Handling Resource Sharing. Mobile Operating Systems – Architecture – Layers – Microkernel Design – Kernel Extensions – Processes and Threads – Memory Management – File system – Android – iOS.

(12)

Reference Books:

1. Abraham Silberschatz and Peter Baer Galvin, "Operating System Concepts", 8th Edition, Pearson Education, 2009.
2. Mukesh Singhal and Niranjan G. Shivaratri, "Advanced Concepts in Operating Systems – Distributed Database, and Multiprocessor Operating Systems", Tata McGraw-Hill, 2017.
3. Andrew S.Tanenbaum, "Distributed Operating Systems", Pearson Education, 2013.
4. Rajib Mall, "Real-Time Systems: Theory and Practice", Prentice Hall, 2006.
5. Philip Levis, "TinyOS Programming", Cambridge Publishers, 2009.

23MCS102 THEORY OF COMPUTATION

Total No. of Hours: 60 **Hours per week: L:T:P (3:1:0)**

Course Objectives: This course provides the broad coverage of automata and regular expressions. The students will be equipped with the knowledge about grammars, push down automata and Turing machines.

Learning Outcomes
After completion of this course, the student will be able to:

- Implement programming techniques for Turing Machine
- Understand language of Pushdown Automata
- Draw parse trees.

UNIT I: Automata and Regular Expressions
Introduction to formal proof – Additional forms of proof – Inductive proofs –Finite Automata (FA) – Deterministic Finite Automata (DFA) – Non-deterministic Finite Automata (NDFA) – Finite Automata with Epsilon transitions.
Regular Expression – FA and Regular Expressions – Proving languages not to be regular – Closure properties of regular languages – Equivalence and minimization of Automata. **(12)**

UNIT II: Grammars
Grammar Introduction– Types of Grammar - Context Free Grammars and Languages— Parse Trees – Ambiguity in grammars and languages – Simplification of CFG – Elimination of Useless symbols - Unit productions - Null productions – Normal forms for CFG- Greiback Normal form – Chomsky normal form – Pumping Lemma for CFL. **(12)**

UNIT III: Push Down Automata
Definition of the Pushdown automata – Languages of a Pushdown Automata – Equivalence of Pushdown automata and CFG– Deterministic Pushdown Automata. **(12)**

Unit IV: Turing Machines
Turing Machines, Introduction – Formal definition of Turing machines – Instantaneous descriptions- Turing Machine as Acceptors – Turing Machine as Transducers- Programming Techniques for Turing Machines– Modifications of Turing Machines. **(12)**

Unit V - Computational Complexity
Undecidability - Basic definitions- Decidable and undecidable problems - Properties of Recursive and Recursively enumerable languages – Undecidable problems about Turing Machine – Introduction to NP-Hardness and NP-Completeness. **(12)**

Reference Books:

1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", 3rd Edition, Pearson Education, 2008.
2. H.R. Lewis and C.H. Papadimitriou, "Elements of the theory of Computation", 2nd Edition, Pearson Education, 2003.
3. Thomas A. Sudkamp," An Introduction to the Theory of Computer Science, Languages and Machines", 3rd Edition, Pearson Education, 2007.
4. Raymond Greenlaw an H.James Hoover, "Fundamentals of Theory of Computation, Principles and Practice", Morgan Kaufmann Publishers, 1998.
5. Micheal Sipser, "Introduction of the Theory of Computation", 2nd Edition, Thomson Brokecole, 2006.
6. J. Martin, "Introduction to Languages and the Theory of computation", 3rd Edition, Tata Mc Graw Hill, 2007.

23MCS103 ADVANCED COMPUTER ARCHITECTURE

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: The aim of this course is to understand concepts of basic computer architecture such as register transfer logic and arithmetic operations and multiprocessors.

Learning Outcomes:
After completion of this course, the student will be able to:

- Understand the theory and architecture of central processing unit.
- Analyze some of the design issues in terms of speed, technology, cost, performance.
- Learn different parallel computer methods.

UNIT I: Basic Computer Organization and Design: Structure of Computers: Computer types, Functional units, Basic operational concepts, Von- Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer. Instruction codes, Computer Registers, Computer Instructions and Instruction cycle. Timing and Control, Memory-Reference Instructions, Input/output and interrupt.
Central processing unit: Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC), CISC v/s RISC. **(14)**

UNIT II: Register Transfer and Micro-operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro Operations, Arithmetic logic shift unit. Micro-programmed Control: Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit. Input Output: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. Instruction level parallelism: Instruction level parallelism (ILP)-over coming data hazards, limitations of ILP. **(12)**

UNIT III: Processors and Memory Hierarchy: Advanced processor technology - Super scalar and vector processors - Memory hierarchy technology - Virtual memory technology. Bus, Cache and Shared Memory: Bus System-Cache memory organizations-Shared memory organization-Sequential and weak consistency models. **(12)**

UNIT IV: Parallel Computer methods: Multiprocessor and multi computers – Shared-Memory multiprocessors, Distributed-Memory Multiprocessors. Multi-vector and SIMD computers. PRAM and VLSI models - Architectural development tracks - Multiple Processor Tracks, Multi- vector and SIMD Tracks, Multi-threaded and Dataflow Tracks. Thread level parallelism: Multi-threaded Architectures, Distributed Memory MIMD Architectures, and Shared Memory Architectures. **(12)**

UNIT V: Scalable, Multithreaded and Dataflow Architectures: Principles of multithreading, Scalable and Multithreaded Architectures- The Standford Dash Multiprocessor - The Kendall Square Research KSR- 1 -Tera Multiprocessor System, Dataflow and Hybrid Architectures – Evolution of Dataflow Computers – The ETL/EM4 in Japan, The MIT/Motorolo *T Prototype. **(10)**

Reference Books:

1. Mano M Morris, "Computer System Architecture", 3rd Edition, Pearson India, 2019.
2. William Stallings, "Computer Organization and Architecture designing for performance", 10th Edition, Pearson India, 2016.
3. Kai Hwang, "Advanced Computer Architecture – Parallelism, Scalability, Programmability", 3rd Edition, Tata McGraw-Hill, 2017.
4. Dezso Sima, Terence Fountain, Peter Kacsuk, "Advanced Computer Architectures – A Design Space approach", Pearson Education, 2009.
5. John L. Hennessy, David A. Patterson Berkeley "Computer Architecture A Quantitative Approach", 6th Edition, Morgan Kaufmann, 2019.

23MCS104 DATA STRUCTURES

Total No. of Hours: 60 +60 **Hours per week: L: T: P (4:0:4)**

Course Objectives: Computer science is primarily concerned with the study of data. It is important to introduce the student to these aspects of data and data structures which are required in modular programming. The basic algorithms related to handle data like stack, queue, tree and graphs are introduced in this subject.

Learning Outcomes:

After completion of this course, the student will be able to:

- Understand basic data structures such as arrays, linked lists, stacks and queues.
- Ability to program data structures and use them in implementations of abstract data types.
- Describe the hash function and concepts of collision and its resolution methods
- Solve problem involving graphs, trees
- Apply algorithm for solving problems like sorting, searching, insertion and deletion of data
- Ability to sensibly select appropriate data structures for problems and to justify that choice.

UNIT I: Introduction - Definition of Data, data objects, data types, data structure. Purpose of a data structure and implementation of data structure. Introduction to algorithms, properties of algorithms.
Arrays: Array as data structure, Storage representation of arrays, polynomial representation using arrays, sparse matrix representation. Applications of arrays. **(12)**

UNIT II: Searching Techniques – linear and binary search. Hashing, hash functions, collision and collision resolution.
Sorting – insertion sorts, selection sorts, bubble sorts, quick sort, radix sort, enumeration sort and merge sort. **(12)**

UNIT III: Lists: Linear list concepts – single and doubly linked list, circular lists, applications, operations on linked list.
Stacks: Operations on stack, static and dynamic implementation, application of stack: recursion, prefix, infix, and postfix expressions. **(12)**

UNIT IV: Queues – Operations on Queue, static and dynamic implementation, Types of Queue – circular, priority, dequeue. Application of Queue: job scheduling.
Trees: Tree terminology, Binary tree, Binary tree representation, introduction to binary search tree (BST). Operation on BST: creation, traversal, insertion and deletion in binary tree, binary tree sort. **(14)**

UNIT V: Balanced Tree: AVL tree, height balance in AVL trees. B-tree, insertion and deletion into a B-tree.
Graphs: Terminology, representation, traversal, operations and applications: single source shortest path problem, minimum spanning tree. **(10)**

Suggested List of Lab Exercises:

1. Sorting programs: Bubble sort, Merge sort, Insertion sort, Selection sort, and Quick sort.
2. Searching programs: Linear Search, Binary Search.
3. Array implementation of Stack, Queue, Circular Queue, Linked List.
4. Implementation of Stack, Queue, Circular Queue, Linked List using dynamic memory allocation.
5. Evaluation of arithmetic expression
6. Implementation of Binary tree.
7. Program for Tree Traversals (preorder, inorder, postorder).
8. Program for graph traversal (BFS, DFS).
9. Write a program to perform the following operations:
   a) Insert an element into an AVL tree.
   b) Delete an element from an AVL tree.
   c) Search for a key element in an AVL tree.

Reference Books:

1. Robert L. Kruse, Bruce P. Leung, Clovis L. Tondo, "Data structures and program design in C", BPB Publications, 2010.
2. Mark Allen Weiss, "Data structures and Algorithm Analysis in C", Tata McGraw hill Publishing, 2nd edition, 2014.
3. Jean Paul Trembly and Soberson, "Introduction to Data Structures", Tata McGraw hill Publishing, 2017.
4. Gilberg R.F, Forouzan B.A and Cengage, "Data Structures: A pseudocode approach with C++", 2nd edition, 2005.
5. Dinesh P. Mehta, Sartaj Sahni, "Handbook of Data Structures and Applications", Second Edition, CRC Press, 2018.

23MCA105 OBJECT ORIENTED CONCEPTS USING JAVA

Total No. of Hours: 60 + 60 **Hours per week: L:T:P(4:0:4)**

Course Objectives: The objective of this paper is to teach the fundamentals of the Java language. Java is a pure object-oriented language, language of the Internet.

Learning Outcomes:
After completion of this course, the student will be able to:

- Design problem solutions using Object Oriented Techniques
- Apply the concepts of data abstraction, encapsulation and inheritance for problem solutions
- Write and execute Java programs based on collection framework and hibernate

UNIT I: Introduction to object-oriented programming: procedural approach Vs object oriented approach, principles of OOP: encapsulation, inheritance and polymorphism. Concepts of OOP: Abstraction, overloading, reusability, extensibility, dynamic binding, message passing. Introduction to Java: Importance and features of java, data types, Operators and expressions, Control statements. Introducing classes, objects and methods: defining a class, adding variables and methods, creating objects, constructors. Arrays and strings: creating an array, one and two dimensional arrays, string array and methods, String and String Buffer classes. **(12)**

UNIT II: Inheritance: Basics types, using super, Multilevel hierarchy, abstract and final classes. Using super - Method overriding - Dynamic Method Dispatch, Abstract keyword Using final with inheritance - the Object Class. Packages and interfaces: Basics, Access protection, Extending Interfaces and packages. CLASSPATH variable Exception Handling: Fundamentals, exception types, uncaught exceptions, try-catch block, throw, throws, finally, built in exceptions, creating your own exceptions **(12)**

UNIT III: Multithreaded Programming: Fundamentals, single and multiple thread creation, priorities, synchronization, Thread class, Thread class methods, Runnable interface, inter thread Communication, suspending, resuming and stopping threads. The Collections Framework, Collections Overview, Collection Interface, List Interface, Set Interface, SortedSet Interface, Queue Interface, Java Map Interface. ArrayList Class, LinkedList Class, HashSet Class, Using an Iterator – The For Each Statement. **(12)**

UNIT IV: Introduction to HIBERNATE - Advantages of Hibernate compared to JDBC, ORM (Object Relational Mapping) –Introduction to HIBERNATE Resources - Configuration File -Mapping File -Persistent Class Or POJO -Client Application Hibernate Architecture -Hibernate Architecture -Installation And Directory Structure - Installation And Directory Structure -Hibernate Data Types -Hibernate Data Types. Hibernate API - Configuration -Sessionfactory -Session –Transaction. Object Lifecycle In Hibernate - Transient Object -Persistent Object -Detached Object Crud Operations Using Session Methods -Save, Persist, Save or update, Update, Merge, Delete,Load, Get. **(12)**

UNIT V: Servlets: Servlets Basics – Life Cycle of a Servlet –The Servlet API – Servlet Interfaces – Generic Servlet ClassHttpServletRequest Interface – HttpServeltResponse Interface – HttpServet Class – The Cookie Class – Handling HTTP GET Request – Handling HTTP POST Request Spring Core - Introduction to Spring Frameworks –What is a Core Container –Introduction to IOC –Types of DI –Setter DI vs Constructor DI –Bean Scopes. **(12)**

Suggested List of Lab Exercises:

Implementation of the following OOPs and Java programming concepts learnt in theory.

- Class and objects
- Constructor
- Arrays (Single dimensional and Two Dimensional)
- String manipulation
- Inheritance-Different types of inheritance
- Abstract and final class
- Method overriding
- Packages
- Interfaces
- Exception handling
- Multithreading
- The Collection Framework
- Hibernate-CRUD operations using session methods
- Servlet

Reference Books:

1. Patrick Naughton and HerbertzSchildt, "Java-2 The complete Reference", 12th Edition, 2020.
2. Richard A. Johnson, "An Introduction to Java Programming and Object Oriented Application Development", CENGAGE Learning India Pvt. Ltd., New Delhi, 7th Edition, 2016.
3. E. Balagurusamy, "Programming with Java: A Primer", McGraw-Hill Education (India), New Delhi, 6th Edition, 2019.
4. Harvey M. Deitel & Paul J. Deitel, "Java: How to Program", PHI Learning Pvt. Ltd., New Delhi, 11th Edition, 2014.
5. Dr. R. Nageswara Rao, "Core Java An Integrated Approach", Dreamtech Press (India) Pvt. Ltd., Hyderabad, 2nd Edition, 2015.

23MCS106 PYTHON PROGRAMMING

Total No. of Hours: 60 **Hours per week: L:T:P (0:0:4)**

Course Objectives:
This course introduces the student to Python programming language through its core language basics and program design techniques suitable for modern applications.
To understand the wide range of programming facilities available in Python and to utilize high-performance programming constructs to develop solutions in real life scenarios.

Learning Outcomes:
After completion of this course, the student will be able to:

- To understand the wide range of programming facilities available in Python and develop solutions to simple computational problems
- To utilize high-performance programming constructs available in Python to represent and process compound data
- To develop solutions for real-time projects in areas that includes data analytics, visualization, image processing, artificial intelligence and machine learning

Topics Covered:

- Conditional and looping constructs, Functions
- Modules - Functions from math, random, time& date module.
- Strings: Regular Expressions and Pattern matching
- List – Accessing List Elements, List operations, List comprehensions
- Dictionaries: Traversing, appending, updating and deleting elements
- Tuples
- I/O and File Handling – Working with CSV files and other file formats
- Arrays and Matrices – Working with NumPy Module
- Object Oriented Concepts - Python Scopes and Namespaces, Inheritance
- Working with Pandas - Data indexing, selection, Operating on Data
- Creating simple 2D and 3D plots using Matplotlib

Reference Books:

1. Paul Gries, Jennifer Campbell, Jason Montojo, “Practical Programming: An introduction to Computer Science Using Python”, 2nd Edition, The Pragmatic Bookshelf, 2013.
2. Allen Downey, Jeffrey Elkner, “Learning with Python: How to Think Like a Computer Scientist”, Createspace Independent Publishers, 2015.
3. Hans Fangohr, “Introduction to Python for Computational Science and Engineering (A beginner's guide)”, 2015.
4. Timothy A. Budd, “Exploring Python”, Mc Graw Hill Education, 2011.
5. Mark Lutz, “Learning Python”, 4th Edition, O’Reilly publication, 2012.

Semester II

23MCS201 ARTIFICIAL INTELLIGENCE

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives:
The objective of the paper is teach modern view of AI and its foundation. It illustrates search strategies with algorithms and problems. It teaches different techniques of NLP and Game Playing.

Learning Outcomes:
- Identify problems that are amenable to solution by AI methods.
- Identify appropriate AI methods to solve a given problem
- Formalize a given problem in the language/framework of different AI methods
- Implement basic AI algorithms
- Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports

UNIT I: Introduction to AI - The Foundations of AI, AI Technique -Tic-Tac-Toe. Problem characteristics, Production system characteristics, Production systems: 8-puzzle problem. Searching: Uniformed search strategies – Breadth first search, depth first search.
Local Search Algorithms: Generate and Test, Hill climbing, simulated annealing search, Constraint satisfaction problems, Greedy best first search, A* search, AO* search. **(12)**

UNIT II: Game Playing Overview, Minimax algorithm, Alpha-Beta pruning, Additional Refinements. Probabilistic Reasoning: Ad Hoc Methods. **(8)**

UNIT III: Knowledge-based agents; Logic: propositional logic reasoning patterns in propositional logic; Effective propositional inference; Agents based on propositional logic. Representation revisited; Syntax and semantics of first-order logic; Using first-order logic; Knowledge engineering in first-order logic. Propositional versus first-order inference; Unification and lifting, Forward chaining; Backward chaining; Resolution, Truth maintenance systems. **(14)**

UNIT IV: Basic plan generation systems – Strips -Advanced plan generation systems – Kstrips – Strategic explanations -Why, Why not and how explanations. Learning: Forms of Learning; Inductive learning; Learning decision trees; Ensemble learning; Computational learning theory. Handling Uncertainties: Non-monotonic reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic. **(14)**

UNIT V: Computer Vision, Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking.

Expert Systems: Need and justification for expert systems, Architecture and role of expert systems, Case studies: MYCIN **(12)**

Reference Books:

1. E. Rich and K. Knight, "Artificial Intelligence", 3rd Edition. New York: TMH, 2019.
2. S. Russell and P. Norvig, "Artificial Intelligence A Modern Approach", 4th Edition. Pearson Education, 2020.
3. Andries P. Engelbrecht, "Computational Intelligence: An Introduction", John Wiley & Sons, 2nd Edition, 2009.
4. Eugene Charniak and Drew McDermott, "Introduction to Artificial Intelligence", 2nd Edition, Pearson Education, 2009.
5. George F Luger, "Artificial Intelligence Structures and Strategies for Complex Problem Solving", 6th Edition. Pearson Education, 2009.

For Continuous Internal Assessment Only

Implementation of basic algorithms learnt in the theory.

23MCS202 ADVANCED COMPUTER NETWORKS

Total No. of Hours: 60 **Hours per week: L:T:P(4:0:0)**

Course Objectives: This course provides the broad coverage of data communication system and computer networks. The students will be equipped with the knowledge about communication system components, internetworking, network topology, protocols, and algorithms.

Learning Outcomes

After completion of this course, the student will be able to:

- Understand the concept of networks, different topologies and network devices
- Discuss the objectives and functionalities of different layers.
- Describe how the available methods and algorithms are implemented in the real-time networks such as Ethernet, Bluetooth and internet protocols.
- Understand the working of few application protocols such as SMTP, POP and HTTP.

UNIT I: Introduction - Protocol and Standards - Hierarchies, Network Models, Layered Tasks - OSI Reference Model, Introduction to internetworking, TCP/IP Model.

Data Communication: Analog and Digital Signals, Digital Transmission – Line Coding, Block Coding, Sampling and Transmission Mode, Modulation Techniques, Networking Devices – hubs, switches, bridges, routers and gateways, Bandwidth utilization - Multiplexing and Spreading, Switching Techniques, Packet switching protocols, Introduction to X.25. **(12)**

UNIT II: Optical Networking – Introduction, SONET / SDH Standard, WAN protocol – Introduction to ATM, basic concepts of ATM Networking, ATM cells.
LAN Protocol Architecture – Wireless LAN's, Bluetooth, and High-Speed LAN's.
(12)

UNIT III: Internet and Transport Protocols **-** Internet basics, IP Protocol, Introduction to Intra-domain and inter-domain routings, Unicast Routing Protocols, Multicast Routing Protocols, Overview of RIP, OSPF, MOSPF and BGP protocols, Transport protocols - TCP, UDP protocols. **(12)**

UNIT IV: Software Defined Networking: Evolution of Switches and Control Planes, cost, Fundamental characteristics, operations, SDN devices, SDN controller, Openflow Specification: Openflow overview, OpenFlow 1.0 and 1.1, Introduction to Network Functions Virtualization, SDN vs NFV.
(12)

UNIT V: Network Security: Security Services, Message Confidentiality – Symmetric Key Cryptography, AES algorithm, Public Key cryptography, RSA algorithm, Message Integrity – hashing, SHA algorithm, MAC, Digital Signature, Entity Authentication, Key Management – KDC, Diffie Hellman and Kerberos. **(12)**

Reference Books:

1. Andrew S. Tanenbaum, "Computer Networks", Sixth Edition, Pearson Education, 2021.
2. Behrouz A. Forouzan, "Data communications and Networking", Tata McGraw-Hill, Fourth Edition, 2017.
3. Stallings William, "Data & Computer Communications", Pearson Education Asia, Tenth Edition, 2017.
4. Paul Göransson, Chuck Black, "Software Defined Networks, A Comprehensive Approach", 2nd Edition, 2016, Elsevier, ISBN: 978-0-12-416675-2
5. William Stallings, "Network Security Essentials", 6th edition, Pearson, 2017.
6. Dayanand Ambawade, Deven Shah, Mahendra Mehra, Mayank Agarwal, "Advanced Computer Network", Wiley Publications, 2011.

For Continuous Internal Assessment Only

Implementation of Computer Networking concepts such as Packet Capture and Analysis, Secure Communication Exercise, Network Vulnerability Assessment and Network troubleshooting using Cryptool / Packet Tracer / Similar tools.

23MCS204 ADVANCED DATABASE MANAGEMENT SYSTEM

Total No. of Hours: 60 + 60 **Hours per week: L:T:P (4:0:4)**

Course Objectives:

- To provide overview of advancement in SQL and Database technology.
- To impart knowledge of query processing and optimization.
- To introduce the concept of document-oriented database.
- Understand the usage of advanced data models for real life application.

Learning Outcomes

On completion of this course, the student will be able to:

- Discuss new developments in database technology.
- Measure query cost and optimize query execution.
- Demonstrate the understanding of the concepts of document-oriented databases.
- Implement advanced data models for real life applications.

UNIT I: Introduction: Purpose of Database Systems - View of Data - Database Languages - Data Storage and Querying - Database Users and Administrators. Database System Architecture: Database System Architectures, Distributed Databases, Parallel Databases. Data Models: Entity-Relationship Model, Relational Model.

Relational Databases: Introduction to the Relational Model - Structure of Relational Databases - Database Schema - Keys-Schema Diagrams - Functional Dependency – Normalization. Relational Query Languages - Relational operations. Advanced SQL - Accessing SQL from a Programming Language – Triggers. **(12)**

UNIT II: Database Storage: File organization, Organization of records in files, Data Dictionary storage.
Indexing and Hashing: Basic Concepts, Ordered Indices, B+-Tree Index Files, Static Hashing, Dynamic Hashing. **(12)**

UNIT III: Transaction Processing and Concurrency Control: Definition of Transaction and ACID properties; Concurrency Control Techniques: Lock based Concurrency control - Optimistic Concurrency Control – Timestamp based Concurrency Control, Deadlock Handling. **(12)**

UNIT IV: Document Oriented Database: Need of Document Oriented database, Types of encoding - XML, JSON, BSON, Representation of JSON Objects.
Introduction to MongoDB, Storing data and accessing data from MongoDB, Querying MongoDB, Document store internals, MongoDB reliability and durability, Horizontal scaling, CRUD operations in MongoDB, Creating and using indexes in MongoDB.
(12)

UNIT V: Advanced data models: Temporal data models: Aspects of valid time, bi-temporal time and bi-temporal time with examples of each.
Spatial model: Types of spatial data models - Raster, Vector and Image models.

(12)

Suggested List of Lab Exercises:

- Data modelling - E-R & Relational model
- Relational data model – working with SQL: DDL & DML commands
- Aggregate functions, Sub queries and Joins
- Concept of a view – Creation of views, operations
- Working with Stored Procedures, Triggers, Functions, and Cursors.
- Simulation of Query optimization
- Data handing using JSON
- CRUD operations in MongoDB
- Working with images and videos in MongoDB
- Processing of Spatial and temporal data

Reference Books:

1. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Addison Wesley, Pearson Education, Seventh Edition, 2021.
2. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Tata McGraw Hill, Sixth Edition, 2013.
3. Jeff A Hoffer, V Ramesh, HeikkiTopi, "Modern Database Management", Global Education, Twelfth Edition, 2016.
4. Kristina Chodorow, MongoDB, "The definitive Guide", O'Reilly, Third Edition, 2019.
5. Raghu Ramkrishnan, Johannes Gehrke, "Database Management System" Tata McGraw Hill. Third edition, 2020.

23MCS205 WEB TECHNOLOGIES

Total No. of Hours: 60 + 60 **Hours per week: L:T:P (4:0:4)**

Course Objectives: This course aims to familiarize students with cutting-edge technologies that facilitate the development of intricate web applications. Participants will explore advanced JavaScript concepts for creating interactive web applications with sophisticated user interfaces. Additionally, they will delve into server-side web technologies that power dynamic web applications. The course also covers the creation of contemporary web applications using React and Node frameworks.

Learning Outcomes:

- Understand the foundations of web development and the client-server architecture.
- Create well-structured HTML documents and apply CSS styles for web design.
- Apply JavaScript effectively to create interactive and dynamic websites.
- Develop dynamic web applications using the React framework, including components, services, and routing.
- Demonstrate proficiency in developing and deploying a web application using React.js and Node.js.

Unit I: Basic Web Concepts: Introduction to World Wide Web (WWW), Web Browsers and Web Servers, Security and Vulnerability in Web Systems, Web System Architecture – URL and Domain Name, Client-side and Server-side Scripting, Web hosting options and deployment strategies, Web protocols – HTTP Request Response Format, Features of HTTP/1, HTTP/2 and HTTP/3 protocols.
HTML Fundamentals – Elements and attributes, Features of Web 3.0, HTML vs HTML5, Semantic HTML and Accessibility. CSS3.0 – Styles and Style sheets, Selectors, Style properties, Box Model, and Responsive design. **(12)**

Unit II: JavaScript: Introduction, Operators and Expressions, JavaScript Programming Constructs, Document Object Model, Working With Objects, Handling Events, Creating Frames and Windows, Processing Forms, Using Images, Regular expressions, Working with cookies. Introduction to JSON-JSON vs XML-JSON Objects, AJAX: Ajax Client Server Architecture - XML Http Request Object - Call Back Methods. ES6 and modern JavaScript features, Call-backs and Promises, Single Page Application, Asynchronous Communication. **(14)**

Unit III: React.js: React installation and application setup, JSX, React Classes and Components, Rendering of elements, Properties, State, Context, Component lifecycle methods, Refs & Keys, Handling events and forms, React Router, Stateless components, React form & controls, State management with Redux or Context API, Asynchronous data fetching with Axios or Fetch API, React hooks – useState, useEffect. **(12)**

Unit IV: Node.js: Understanding Node.js Architecture, Setting up a Node.js development environment, Working with modules and npm, Handling HTTP requests and responses with Node.js, query string, call backs, buffers, streams, File system, Working with databases – Simple SQL operations using MySQL: Reading and Writing to MySQL. Introduction to Testing frameworks Jest, Mocha, and Chai. Client-side Vs Sever-side rendering. **(12)**

Unit V : Express.js; Introduction to Web services and REST API's , Express Installation and Server setup, Building the application stack, Routing, List API, Create API, Error Handling, Express Middleware, User authentication and authorization – Types of Authentications in Node.js – JWT Token and OAuth, Express Scaffolding and Templates. **(10)**

Suggested List of Lab Exercises:

- Web page creation using HTML5 elements
- Using style sheets (CSS2 and CSS3)
- Client Side Scripting using Java Script
- Working with images, JavaScript objects and event handling
- Form validation using JavaScript, Working with JSON
- Designing webpages with React.js components
- State management with Redux or Context API,
- Asynchronous data fetching with Axios or Fetch API
- React hooks – useState, useEffect.
- Database handling using Node.js & MySQL
- Creating Simple dynamic web application.

Reference Books:

1. Laura Lemay, Rafe Colburn & Jennifer Kyrnin, "Mastering HTML, CSS & Javascript Web Publishing", BPB Publications, First edition,2016.
2. Harvey M Deitel, Paul J Deitel and Tem R Nieto, Internet and World Wide Web How to Program, Pearson, 6th Edition, 2020.
3. Minnick, C. Beginning ReactJS foundations building user interfaces with ReactJS: An Approachable Guide, OReillly, 2022.
4. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer, Apress Publications, 1st Edition, 2018.
5. Brown, Ethan, "Web Development with Node and Express: Leveraging the JavaScript Stack", O'Reilly Media, Second Edition, 2019.
6. "Learning React, Functional Web Development with React and Redux", Alex Banks and Eve Porcello, O'Reilly Media, May 2017.

23MCS206 AGILE SOFTWARE ENGINEERING

Total No. of Hours: 60 **Hours per week: L:T:P (0:0:4)**

Course Objectives: This course introduces candidates to agile way of software development. This course takes the candidate through the software engineering lifecycle with an agile flavor. It introduces them to breaking down requirements, delivering minimum viable product each sprint, best practices to maintain code, importance for automation and tools needed to automate the software development process.

Learning Outcomes:

- Ability to apply the concepts learnt in software engineering methodologies theory on Agile projects
- To develop project with clean code
- Ability to understand the integration tools

Topics Covered:

1. Writing user stories
2. Different roles of scrum
3. Team presentation
4. Working with Sonarqube tool
5. Working with JUnit Framework
6. Practices to write clean code
7. Continuous integration using Jenkins tool

.

References Books:

1. Ian Summerville, "Software Engineering", Pearson Education Ltd, 10th Edition, 2015, ISBN: 9780133943030.
2. Roger S Pressman, Bruce R. Maxim, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 8th Edition, 2015, ISBN: 9780078022128.
3. "Guide to the Software Engineering body of Knowledge", Version 3.0 SWEBOK, IEEE Computer Society Press, 2014, E-Book: ISBN: 9780769551661.
4. Pankaj Jalote, "An Integrated Approach to Software Engineering", 3rd Edition, Narosa Publishing House 2013, ISBN: 81-7319-702-4.
5. Ken Schwaber, MikeBeedle, "Agile Software Development with Scrum", Pearson India, 2014.
6. Robert C.Martin, "Clean Code - A Handbook of Agile Software Craftsmanship", Prentice Hall, 2008.

Recommended Reading:

1. The Phoenix Project: A novel about IT, DevOps, and helping your Business Win-Gene Kim, Kevin Behr and George Spafford, It Revolution Press, 2018.

Discipline Specific Electives

ELECTIVE I – AI & ML

23MCS203A DIGITAL IMAGE PROCESSING

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

.

Course Objectives: This course exposes the students to the basics of digital image processing and its applications. It covers a broad range of image processing techniques such as Image enhancement, restoration, segmentation and feature analysis. The course also provides develop on-hand experience in applying these techniques to process the images.

Learning Outcomes

- Have a clear perceptive and practical scope of digital image processing, current technologies and issues that are specific to image processing systems.
- Understand the working of different image processing algorithms such as filtering, segmentation, morphological processing and image representation.
- Could implement basic image processing algorithms using image processing tools such as MATLAB or OpenCV

UNIT I: Fundamentals of Image Processing: Introduction, Components of image processing system, Elements of visual perception, Steps in Image Processing Systems, Image Acquisition, Sampling and Quantization, Pixel Relationships, Colour Fundamentals and RGB, CMY, HSV colour Models.
Mathematical Preliminaries: Vector algebra, Matrix operations, Fourier Transform. **(12)**

UNIT II: Image Enhancement: Introduction to Spatial and Frequency domain, Image Operations, Arithmetic, Logical, Statistical and Spatial Operations, Convolution and Correlation, Enhancement in Spatial Domain – Gray level Transformations, Histogram Processing, Spatial Filtering, Smoothing and Sharpening. **(12)**

UNIT III: Filtering in Frequency Domain: Smoothing and Sharpening filters, Homomorphic Filtering.
Image Restoration: Noise models, Constrained and Unconstrained restoration models. **(10)**

UNIT IV: Image Segmentation: Detection of Discontinuities, Edge Operators, Edge Linking and Boundary Detection, Thresholding, Region Based Segmentation.
Feature Analysis and Extraction: Image Features, Textures, Boundary representations and Descriptions, Component Labeling, Regional descriptors and Feature Selection Techniques. **(14)**

UNIT V: Image Morphology: Binary and Gray level morphology operations, Erosion, Dilation, Opening and closing operations, Distance transforms, Basic morphological operations.
Image Compression: Error criterion, Lossy Compression, Lossless Compression, Huffman Coding, Run length Coding, Block Coding, Quad Tree Coding, contour Coding. . **(12)**

Reference Books:

1. Rafael C.Gonzalez and Richard E.Woods, "Digital Image Processing", Third Edition, Pearson Education, 2008.
2. S. Sridhar, "Digital Image Processing", Oxford University Press, 2011.
3. Jayaraman S, Veerakumar T, Esakkirajan S, "Digital Image Processing", Tata McGraw-Hill Education, 2011.
4. Rafael C. Gonzalez, Richard E. Woods , Steven L. Eddins ,"Digital Image Processing Using Matlab 2E", Gatesmark Publishing, 2009.

For Continuous Internal Assessment Only

Implementation of image processing concepts such as enhancement, restoration, segmentation and compression using Matlab / ImageJ / OpenCV / Similar tools

ELECTIVE I – CYBER SECURITY

23MCS203A WEB APPLICATION SECURITY

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

.

Course Objectives: This course provides students basic knowledge and skills in detecting and defending threat to web application.

Learning Outcomes: On completion of this course, the student will be able to:

- State the need and scope for cyber laws
- Enumerate various web attacks, and describe their sources, and mechanisms of prevention
- Identity various threats associated with a web application and helps in building a secure web application
- Provide countermeasures against authentication and input injection attacks

UNIT I: Introduction: Cyberspace, Cyber Crimes, Cyber criminals, Cyber security, Cyber Security Threats, Cyber laws and legislation, Law Enforcement Roles and Responses**,** The Indian IT ACT 2000 and amendments. Introduction to SCADA (supervisory control and data acquisition) Understanding SCADA security policies.

Web Application Hacking and Profiling: Introduction to Web Application Hacking - GUI web Hacking, URI Hacking, HTTP Methods Headers and Body, The Web Client and HTML, Other Protocols, How & Why Web Applications are attacked, Social Engineering, OWASP Top 10 security risks.
Infrastructure Profiling – Footprinting and Scanning, Basic Banner Grabbing, Advanced HTTP Fingerprinting, Infrastructure Intermediaries. Application Profiling – Manual Inspection, Search Tools for Profiling, Automated Web Crawling, General Countermeasures. **(14)**

Unit II: Web Authentication Threats: Username/password Threats, Password Guessing and its Countermeasures, Eavesdropping attacks and its Countermeasures, Forms-based Authentication attacks and its countermeasures. Stronger web Authentication, Web Authentication Services. Secure authentication – Introduction to Multi Factor Authentication (MFA), SAML, OAuth, Bypassing Authentication: Token Replay, CAPTCHA Bypass, Cross-site Request Forgery, and Identity Management. **(10)**

Unit III: Input Validation and Fuzzing: Attack vectors, Common Input Injection Attacks: Buffer Overflow, Canonicalization and its countermeasures, Advanced Directory Traversal, Navigating without Directory Listing, HTML Injection: XSS, Embedded scripts, Cookies and Predefined Headers, Counter countermeasures. SQL Injection: SUB Queries, UNION, SQL Injection countermeasures, XPATH Injection and its countermeasures, LDAP Injection. **(12)**

Unit IV: Basics of Penetration Testing: The Phase of PTES, Types of Penetration Tests. Metasploit: Introduction, Metasploit Basics: Terminology, and Interfaces.
Phishing and Identity Theft: Introduction, Phishing – Methods of Phishing, Phishing Techniques, Phishing Toolkits and Spy Phishing. Identity Theft – Personally identifiable information (PII), Types of Identity Theft, Techniques of Identity Theft. **(12)**

Unit V: Attacking Users: Defacing Content, Capturing User Input: Using Focus Event, Using Keyboard Events, Using Mouse and Pointer Events, Using Form Events, Social Engineering: Using TabNabbing, Abusing UI Expectations: Using Fake Login Prompts, Pretty Theft, Gmail Phishing. **(12)**

Reference Books:

1. Peter W. Singer and Allan Friedman, Cybersecurity and Cyberwar, Oxford University Press, 2014
2. Jonathan Clough, Principles of Cybercrime, Cambridge University Press, 2015
3. Hacking Exposed Web Application, 3rd Edition by Joel Scambray, Vincent Liu, Caleb Sima
4. 2. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws by Dafydd Stuttard and Marcus Pinto Wiley Publication
5. Metasploit - The Penetration Tester's Guide by David Kennedy , Jim O'gorman , Devon Kearns and Mati Aharoni – No Starch Press Publication
6. The Browser Hacker’s Handbook by Wade Alcorn, Christian Frichot and Michele Orru – Wiley Publication
7. Web Penetration Testing with Kali Linux by Joseph Muniz, Aamir Lakhan – Packt Publication

ELECTIVE I – DATA SCIENCE

23MCS203C STATISTICAL CONCEPTS FOR DATA SCIENCE

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives:
This course is intended to study the basics of Blockchain technology. During this course learner will explore various aspects of Blockchain technology like application in various domains and by implementing learner will have idea about the working of private and public Blockchain, and smart contract.

Learning Outcomes:
After the completion of this course, student will be able to

- To differentiate among kinds of data and know various ways to present them.
- To learn the distributions to perform analysis of various kinds of data.
- Infer the concept of correlation and regression for relating two or more related variables.
- Demonstrate the probabilities for various events.

UNIT I: Organization of Data & Descriptive Statistics: Origin and development of Statistics, Types of data: primary, secondary, quantitative and qualitative data. Types of Measurements: nominal, ordinal, discrete and continuous data. Presentation of data by tables: construction of frequency distributions for discrete and continuous data, graphical representation of a frequency distribution by histogram and frequency polygon, cumulative frequency distributions. **(12)**

UNIT II: Representation of Data and Sampling Distribution: Measures of location or central tendency: Arithmetic mean, Median, Mode, Geometric mean, Harmonic mean. Partition values: Quartiles, Deciles and percentiles. Measures of dispersion: Mean deviation, Quartile deviation, Standard deviation, Coefficient of variation. Moments: measures of skewness, Kurtosis. Random sampling and sample bias, selection bias, Sampling Distribution of a statistic, Confidence Intervals, Normal Distribution, Binomial Distribution, Poisson distribution. **(14)**

UNIT III: Statistical Experiments and Significance Testing: Hypothesis test, Resampling. Statistical Significance and P-values, t-tests, ANOVA, Chi square test. **(12)**

UNIT IV: Correlation And Regression: Correlation: Scatter plot, Karl Pearson coefficient of correlation, and Spearman's rank correlation coefficient. Regression: Concept of errors, Principles of Least Square, Simple linear regression and its properties, Multiple Linear Regression, Polynomial and Spline Regression. **(10)**

UNIT V: Probability theory: Sample Spaces- Events - Laws of total probability- Axioms – Counting - Conditional Probability- Bayes' theorem and its applications, Theorems on probability. **(12)**

Reference Books:

1. Peter Bruce and Andrew Bruce, "Practical Statistics for data Scientists", O'Reilly Publications.
2. Rohatgi V.K and Saleh E, "An Introduction to Probability and Statistics", 3rd edition, John Wiley &Sons Inc., New Jersey, 2015.
3. Gupta S.C and Kapoor V.K, "Fundamentals of Mathematical Statistics", 11th edition, Sultan Chand& Sons, New Delhi, 2014.
4. Mukhopadhyay P, "Mathematical Statistics", Books and Allied (P) Ltd, Kolkata, 2015.
5. Walpole R.E, Myers R.H, and Myers S.L, "Probability and Statistics for Engineers and Scientists", Pearson, New Delhi, 2017.

Open Elective

OPEN ELECTIVE: CYBER SECURITY

Total No. of Hours: 60 **Hours per week: L:T:P(3:1:0)**

Course Objectives: The proliferation of Internet has impacted the lives of people in all professions. Equally, they are also prone to get attacked by hackers and intruders and eventually lose their privacy. The objective of this course to understand the need for Cyber security and its related threats and attacks; Learn methods to become secure in the cyber world and securely communicate in the cyber world and become knowledgeable about the best practices related to cyber security.

Learning Outcomes:
After learning this paper, the students will understand the vulnerabilities of their online presence.
They will also know the different threats and attacks that are possible online and will follow the best practises to counter them and be safe online.

UNIT I: Basics of internet, www, http, html, DNS, IP Address, electronic mail, web browsers, search engines. Social Media: Twitter, Facebook, YouTube, WhatsApp, LinkedIn, advantages, disadvantages.

(12)

UNIT II: Need For Cyber Security: Introduction to security, CIA triad, Case studies, security attacks, privacy and security issues related to social networking, Guidelines Methods to secure oneself in the cyber world, Reversible and Irreversible Cryptographic mechanisms, Applications of Digital Signature, Good password practices

(12)

UNIT III: Secure Transactions: e-commerce, advantages of e-commerce, Online banking security- Online shopping fraud, Guidelines and Recommendations, survey on popular ecommerce sites. Introduction to e-governance, stages of e-governance, advantages, challenges, International Status, Indian status.

(12)

UNIT IV: Everyday Security: Connecting your laptop, mobile devices, PDAs to Internet, Managing your browser, Social Media Security, E-mail security, Safe guarding from Viruses: Antiviruses, Best practices and guidelines.

(12)

UNIT V: Cyber Security Laws: Indian IT Act, 2008 salient features, what is Cyber Forensics?, Functions of cybercrime cell, responding to a cyber-attack.

(12)

Reference Books:

1. Information Technology Amended Act, 2008, Ministry of Law and Justice, Government of India. (deity.gov.in/sites/upload_files/dit/.../itact2000/it_amendment_act2008.pdf)
2. "Information Security Awareness Handbook, ISEA, Department of Electronics and Information Technology", Government of India, 2010

3. Srinivas Bhogle, “E-Governance” Chapter III in Selected Readings on Information Technology Management: Contemporary Issues, Information Science reference, Hershey, New York, page no. 40-61.
4. Tom Huskerson. “Social Media, the Good, Bad, and Ugly: Volume. 3”. 2014.