Starex University, Gurugram

School of Computer Science

Course Structure & Syllabus Outline

Master of Computer Application (MCA)

(Postgraduate Program Effective from Year 2020-21)

STAREX UNIVERSITY, GURUGRAM

SCHEME OF STUDIES AND EXAMINATION

Master of Computer Applications

Scheme effective from 2020-21

SEMESTER 1st

S. No.	Course code	Course Title	Hours per week			Total Contact hrs/week	Credit	Examination Schedule (Marks)				Duration of Exam (Hours)
			L	T	P			Mark of Class work	Theory	Practical	Total	
1	0405101	Discrete Mathematics	3	1	0	4	4	25	75		100	3
2	0405102	Advance Database systems	3	1	0	4	4	25	75		100	3
3	0405103	Web Technologies	3	1	0	4	4	25	75		100	3
4	0405104	Fundamental of computer and Emerging Technologies	3	1	0	4	4	25	75		100	3
5	0405105	Design and analysis of Algorithms	3	1	0	4	4	25	75		100	3
6	0405104	Web Technologies Lab	0	0	4	4	2	50		50	100	3
						Total Credits	22	175	375	50	600	

SEMESTER 2

S. No.	Course code	Course Title	Hours per week			Total Contact hrs/week	Credit	Examination Schedule (Marks)				Duration of Exam (Hours)
			L	T	P			Mark of Class work	Theory	Practical	Total	
1	0405201	Automata Theory	3	1	0	4	4	25	75		100	3
2	0405202	Advance Operating system with Unix	3	1	0	4	4	25	75		100	3
3	0405203	Computer Organization and Architecture	3	1	0	4	4	25	75		100	3
4	0405204	Advance Python Programming	3	1	0	4	4	25	75		100	3
5	0405205	Data Warehouse and Data Mining	3	1	0	4	4	25	75		100	3
6	0405204	Advance Python Programming Lab	0	0	4	4	2	50		50	100	3
7	0405202	Advance Operating system with Unix Lab	0	0	4	4	2	50		50	100	3
						Total Credits	22	175	375	50	600	

SEMESTER 3

S. No.	Course code	Course Title	Hours per week			Total Contact hrs/week	Credit	Examination Schedule (Marks)				Duration of Exam (Hours)
			L	T	P			Mark of Class work	Theory	Practical	Total	
1	0405301	Compiler Design	3	1	0	4	4	25	75		100	3
2	0405302	Soft Computing	3	1	0	4	4	25	75		100	3
3	0405303	Machine Learning	3	1	0	4	4	25	75		100	3
4	0405304	Advance Java Programming	3	1	0	4	4	25	75		100	3
5	0405305	Software Testing and Quality Assurance	3	1	0	4	4	25	75		100	3
6	0405304	Advance Java Programming Lab	0	0	4	4	2	50		50	100	3
						Total Credits	22	175	375	50	600	

SEMESTER 4

S. No.	Course code	Course Title	Hours per week			Total Contact hrs/week	Credit	Examination Schedule (Marks)				Duration of Exam (Hours)
			L	T	P			Mark of Class work	Theory	Practical	Total	
1	0405401	Major Project Report	-	-	-	-	24	-	-	-	500	3
						Total Credits	24	-	-	-	500	

Semester 1

Discrete mathematics

<table>
<tr><td>Course code</td><td colspan="5"></td></tr>
<tr><td>Category</td><td colspan="5">Core Course</td></tr>
<tr><td>Course title</td><td colspan="5"><b>Discrete Mathematics</b></td></tr>
<tr><td rowspan="2"><b>Scheme and Credits</b></td><td><b>L</b></td><td><b>T</b></td><td><b>P</b></td><td><b>Credits</b></td><td rowspan="2">Semester-I</td></tr>
<tr><td><b>3</b></td><td><b>1</b></td><td></td><td><b>4</b></td></tr>
<tr><td>Class work</td><td colspan="5">25 Marks</td></tr>
<tr><td>Exam</td><td colspan="5">75 Marks</td></tr>
<tr><td>Total</td><td colspan="5">100 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

UNIT-I:

Introduction to Counting Principles: Set Theory, Functions and Relations, POSETS and Lattices, Permutation and Combination, Probability, Pigeon-hole principle.

UNIT-II

Mathematical Logic: Propositions, connectives, conditionals and biconditionals, well formed formulas, tautologies, equivalence of formulas, duality law, normal forms, inference theory for propositional calculus; predicate calculus: predicates, free and bound variables, inference theory of predicate calculus.

UNIT-III

Growth of Functions: Asymptotic notations, monotonicity, comparison of standard functions - floors and ceilings, polynomials, exponentials, logarithms and factorials, summations: summation formulas and properties, bounding summations, approximation by integrals.

UNIT-IV

Graph Theory: Basic terminology for undirected and directed graphs, multigraphs and weighted graphs, paths and circuits, Eulerian paths and circuits, Hamiltonian paths and circuits, Planar Graphs, Graph Colouring, Cut sets. Trees: Introduction to Trees, Tree terminology, Prefix codes.

Discrete Numeric Functions and Recurrence Relations Discrete Numeric Functions, Generating functions, Recurrence Relations.

Suggested Readings:

1. C.L. Liu & Mohapatra, Elements of Discrete Mathematics, 4th Edition, 2017, McGraw Hill Education.

2. Kenneth H Rosen, Discrete Mathematics and Its Applications, 7th Edition, 2017, McGraw Hill Education.

3. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, 3rd Edition, 2010, Prentice-Hall of India Learning Pvt. Ltd.

4. Thomas Koshy, Discrete Mathematics with Applications, 2012, Elsevier Academic Press. 5. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms, 1988, John Wiley and Sons.

Advanced database system

Course code					
Category	Core Course				
Course title	**Advance Database System**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-I
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I:

Need of DBMS over traditional Data storage mechanisms, Basic DBMS terminologies; Architecture of a DBMS: Data Independence, DBMS Component Structure, DBMS USERS, various DBMS Data Models: Entity Relationship Model, Importance of ERD, Symbols (Entity: Types of Entities, week Entity, Composite Entity, Strong Entity, Attribute: Types of Attribute, Relationship: Type of relationship, Connectivity, Cardinality).

Normalization and its various forms, Functional Dependencies, Multivalued Dependencies, Join Dependencies Database Integrity: Domain, Entity, Referential Integrity Constraints.

UNIT II:

Relational Languages: Relational Algebra, Relational Calculus, Query Execution, optimization and evaluation Plans.

Transaction Management and Concurrency Control techniques, Database Recovery Management Concepts and methods. Introduction and Need of Database Administration and activities of Database administration.

UNIT III:

Parallel Databases : Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems-Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism.

Distributed Database Concepts : Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing.

UNIT IV:

Multidimensional Databases and their uses in data analytics.

Temporal Databases : Introduction to Temporality, Temporal relationships, temporal hierarchies.

Spatial Databases: Spatial data types, spatial relationships, Topological Relationships, Spatial Data Structures and methods of storage.

Big Data : introduction: introduction to NOSQL Databases (Open Source only). Need and usage of XML Databases: XML Data Model – DTD – XML Schema

Suggested Readings:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database System Concept”, Sixth Edition, 2013, McGraw-Hill

2. Bipin C. Desai, “An Introduction to Database System” , Revised Edition, 2012, Galgotia Publications Pvt Ltd-New Delhi Reference Books: 1. Ivan Bayross, “SQL, PL/SQL The Programming Language of Oracle”, 4th Revised Edition,2009, BPB Publications 2. Peter Rob Carlos Coronel, “Database Systems”, Cengage Learning, 8th ed.

3. C.J.Date, A.Kannan, S.Swamynathan, “An Introduction to Database Systems”, 8th Edition, 2006, Pearson Education. –

Web technologies

Course code					
Category	Core Course				
Course title	**Web Technologies**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-I
	3	**1**		**4**	

Class work	25 Marks
Exam	75 Marks
Total	100 Marks
Duration of Exam	03 Hours

UNIT-I

Introduction: Introduction to Networking, TCP/IP, DNS, Internet and its Evolution, World Wide Web, Web 2.0, Web 3.0, network communication protocols (HTTP/HTTPS, SMTP, IMAP, POP, FTP)

client-server architecture, web applications architecture, application and web servers, web clients.

UNIT-II

Front-end Development: Introduction to HTML5, HTML elements, HTML tags, lists, tables, frames, forms,

Basics of XHTML, CSS style sheets, DOM, XML, XSLT

UNIT-III

Client-Side Programming: JavaScript basic syntax, variables & data-types, literals, functions, objects, arrays, built-in objects, event handling, modifying element style, document trees.

Server-Side Programming: Creation of dynamic content, server-side programming using Java Servlets, Web Services, session management, introduction to JSP and server-side scripting, accessing MySQL / Oracle database from front-end.

UNIT-IV

Web Security, Cookies and Authentication: Security threats, Security risks of a website, Web attacks and their prevention, Web security model, Setting, accessing and destroying cookies, Anonymous Access, Authentication by IP address and Domain, Integrated Windows Authentication, Digital signatures, Digital certificates, Firewalls.

Suggested Readings:

1. Jeffery C. Jackson, Web Technologies: A Computer Science Perspective, Pearson Education India, 2007. Department of Computer Science, University of Delhi 33

2. Achyut Godbole and Atul Kahate, Web Technologies: TCP/IP, Web/Java Programming, and Cloud Computing (3rd ed.), McGraw-Hill Education, 2013.

3. Roger S Pressman and David Lowe, Web Engineering: A Practitioner's Approach, TMH, 2017.

4. Mark Pilgrim, HTML5: Up and Running, O’Reilly | Google Press, 2010 .

5. Jim Keogh, J2EE: The Complete Reference, McGraw Hill Education, 2017

Web Technologies Lab

Course code					
Category	Core Course				
Course title	**Web Technologies Lab**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-I
	0	**0**	4	**2**	
Class work	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Practical list should be prepared based on the content of the subject with following guidelines in mind.

1. Entire syllabus should be covered.

2. Practical list should be designed with real life examples.

3. List should be prepared to cover individual concepts and integration of different concepts on real life problems.

Fundamental of computer and emerging technologies

Course code					
Category	Core Course				
Course title	**Fundamental of computer and Emerging Technologies**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-I
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				

Duration of Exam	03 Hours

UNIT I:

Introduction to Computer: Definition, Computer Hardware & Computer Software Components: Hardware – Introduction, Input devices, Output devices, Central Processing Unit, Memory- Primary and Secondary. Software - Introduction, Types – System and Application. Computer Languages: Introduction, Concept of Compiler, Interpreter & Assembler Problem solving concept: Algorithms – Introduction, Definition, Characteristics, Limitations, Conditions in pseudo-code, Loops in pseudo code.

Operating system: Definition, Functions, Types, Classification, Elements of command based and GUI based operating system.

UNIT II:

Computer Network: Overview, Types (LAN, WAN and MAN), Data communication, topologies Internet : Overview, Architecture, Functioning, Basic services like WWW, FTP, Telnet, Gopher etc., Search engines, E-mail, Web Browsers.

Internet of Things (IoT): Definition, Sensors, their types and features, Smart Cities, Industrial Internet of Things.

UNIT III:

Block chain: Introduction, overview, features, limitations and application areas fundamentals of Block Chain.

Crypto currencies: Introduction, Applications and use cases Cloud Computing: It nature and benefits, AWS, Google, Microsoft & IBM Service.

UNIT IV:

Emerging Technologies: Introduction, overview, features, limitations and application areas of Augmented Reality , Virtual Reality, Grid computing, Green computing, Big data analytics, Quantum Computing and Brain Computer Interface.

Suggested Readings:

1. Rajaraman V., “Fundamentals of Computers”, Prentice-Hall of India.

2. Norton P., “Introduction to Computers”, McGraw Hill Education.

3. Goel A., “Computer Fundamentals”, Pearson.

4. Balagurusamy E., “ Fundamentals of Computers”, McGraw Hill

5. Thareja R., “Fundamentals of Computers”, Oxford University Press.

6. Bindra J., “The Tech Whisperer- on Digital Transformation and the Technologies that Enable it ”, Penguin

Design and analysis of Algorithms

Course code					
Category	Core Course				
Course title	**Design and analysis of Algorithms**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-I
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I

Introduction: Algorithms, Analyzing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time.

Advanced Data Structures: Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps, Tries, Skip List.

UNIT II:

Divide and Conquer with Examples Such as Sorting, Matrix Multiplication, Convex Hull and Searching. Greedy Methods with Examples Such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's Algorithms, Single Source Shortest Paths - Dijkstra's and Bellman Ford Algorithms.

UNIT III:

Dynamic Programming with Examples Such as Knapsack. All Pair Shortest Paths – Warshal's and Floyd's Algorithms, Resource Allocation Problem. Backtracking, Branch and Bound with Examples Such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of Subsets.

UNIT IV:

Selected Topics: Algebraic Computation, Fast Fourier Transform, String Matching, Theory of NPCompleteness, Approximation Algorithms and Randomized Algorithms.

Suggested Readings:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, “Introduction to Algorithms”, Printice Hall of India.

2. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",

3. Aho, Hopcraft, Ullman, “The Design and Analysis of Computer Algorithms” Pearson Education, 2008.

4. LEE "Design & Analysis of Algorithms (POD)",McGraw Hill

5. Richard E.Neapolitan "Foundations of Algorithms" Jones & Bartlett Learning

6. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.

7. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006.

8. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997

9. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.

10. Harsh Bhasin,”Algorithm Design and Analysis”,First Edition,Oxford University Press.

11. Gilles Brassard and Paul Bratley,Algorithmics:Theory and Practice,Prentice Hall,1995

<u>**Semester 2nd**</u>

Automata Theory

Course code					
Category	Core Course				
Course title	**Automata Theory**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-II
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I

Finite Automata: Introduction: Set, Power Set, Super Set, Alphabet, languages and grammars, productions and derivation, Deterministic finite automata (DFA), Non- Deterministic finite automata (NDFA), Equivalence of DFA and NDFA, Conversion of NFA to DFA , minimization of finite automata, Finite automata with ϵ- moves, Acceptability of a string by a finite Automata.

Introduction to Machines: Properties and limitations of Finite Automata, Mealy and Moore Machines, Equivalence of Mealy and Moore machines.

UNIT II

Regular Expression: State and prove Arden’s Method, Regular Expressions, Recursive definition of regular expression, Regular expression conversion to Finite Automata and vice versa.

Properties of regular languages: Regular language, pumping lemma for regular sets/languages, Application of regular languages.

UNIT III

Grammars: Chomsky hierarchy of languages, Relation between different types of grammars, Context-free grammar, Derivation tree / Parse tree, Ambiguity in regular grammar and their removal, Reduced Forms: Removal of useless symbols, null and unit productions, Normal Form: Chomsky Normal form(CNF) and Greibach Normal Form(GNF),

Push Down Automata: Introduction to PDA, Deterministic and Non-Deterministic PDA, Design of PDA: Transition table, Transition diagram and acceptability of strings by designed PDA, Pushdown automata (PDA) and equivalence with CFG.

UNIT IV

Turing machines: The basic model for Turing machines (TM), Deterministic and Non- Deterministic Turing machines and their equivalence, Design of Turing Machines: Transition table, Transition diagram and acceptability of strings by designed turing machine. Variants of Turing machines, Halting problem of Turing machine, PCP Problem of Turing Machine, Linear Bounded Automata, TMs as enumerators.
Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Suggested Readings:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.

2. Introduction to the Theory of Computation, Michael Sipser,3rd edition, Cengage Learning.

3. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.

4. Raymond Greenlaw, H. James Hoover, Fundamentals of the Theory of Computation, Principles and Practice, Morgan Kaufmann, 1998.

5.John C. Martin: Introduction to Languages and Automata Theory, 3rd edition, Tata Mcgraw-Hill, 2007

Advance Operating System with Unix

Course code					
Category	Core Course				
Course title	**Advance Operating system with Unix**				
	L	**T**	**P**	**Credit**	

Scheme and Credits				**s**	Semester-II
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I

Introduction Functions of operating systems, Design approaches: layered ,kernel based and virtual machine approach, why advanced operating systems, types of advanced operating systems.

Distributed Operating Systems Architecture of distributed operating systems, system architecture types, issues in distributed operating systems, inherent limitation of distribute systems, distributed mutual exclusion: classification of mutual exclusion algorithms, Lamport’s ,token based algorithm, Suzuki-Kasami’s Broadcast algorithm, Raymond’s Tree based algorithm, Distributed deadlock detection, Distributed file systems, Distributed shared memory, Distributed scheduling.

UNIT-II

Multiprocessor Operating Systems Introduction, structure of multiprocessor operating system, operating system design issues, threads, the test and set instruction, the swap instruction, implementation of the process wait , processor scheduling, reliability and fault tolerance.

UNIT III

Real Time Operating System Introduction to Real time systems and Real Time Operating Systems, Characteristics of Real Time operating Systems, Classification of Real Time Operating Systems, Services, structure, goal and feature of RTOS, architecture of RTOS, micro kernels and monolithic kernels, tasks in RTOS, Performance measures, estimating program runtimes, task assignment, scheduling in RTOS, rate monotonic scheduling, priority inversion, task management, inter task communication, applications of various RTOS.

UNIT IV

Data base operating Systems Introduction to database operating systems, concurrency control: theoretical aspect, distributed database system, concurrency control algorithms.

Case Study: UNIX File system, File and directly related commands in UNIX, VI Editors, Shell Scripting.

Suggested Readings:

1. Maurice J. Bach, Design of the UNIX Operating System, Prentice Hall, 1986.

2. Silberschatz, Galvin and Gagne, Operating Systems concepts , Wiley, 2013.

3. A S Tanenbaum, Maarten Van Steen, Distributed Operating systems, Pearson.2014.

Advance Operating System with Unix Lab

Course code					
Category	Core Course				
Course title	**Unix Lab**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-II
			4	**2**	
Class work	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Contents:

1 Introduction to UNIX File System.

2. File and Directory Related Commands in UNIX.

3. Essential UNIX Commands for working in UNIX environment.

4. I/O Redirection and Piping

5. Introduction to VI Editors.

6. Introduction of Processes in UNIX

7. Communication in UNIX and AWK.

8. Introduction of the concept of Shell Scripting.

9. Decision and Iterative Statements in Shell Scripting.

10. Writing the Shall Scripts for unknown problems.

Computer Organization and Architecture

Course code					
Category	Core Course				
Course title	**Computer Organization and Architecture**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-II
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I

Data representation: Data Types, Complements, Fixed-Point Representation, Conversion of Fractions, Floating-Point Representation, Gray codes, BCD codes, Excess-3 code, Error Detection Codes.

Register Transfer and Microoperations : Register Transfer Language, Register, Bus and Memory Transfers, Shift Microoperations,.

UNIT II

Basic Computer Organization and Design : Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instruction, Input-Output Instruction.

Central Processing Unit : General Register Organization, Stack organization, Instruction Format, Addressing Modes, Data Transfer and Manipulation, Program Control, RISC, CISC.

UNIT III

Pipelining: Basic Concepts of Pipelining, Throughput and Speedup, Pipeline Hazards.

Parallel Processors: Introduction to Parallel Processors, Locality of reference principle

UNIT IV

Input-output Organization : I/O device interface, I/O transfers–program controlled, interrupt driven and DMA, Software Interrupts.

Memory organization: Memory Hierarchy, Main Memory, **Auxiliary** Memory, Associative Memory, Cache Memory, Associative Mapping, Direct Mapping, Set-Associative Mapping, Virtual Memory.

Suggested books:

1) “Computer System Architecture”, 3rd Edition by M.Morris Mano, Pearson.
2) “Computer Organization and Design: The Hardware/Software Interface”, 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
3) “Computer Organization and Embedded Systems”, 6th Edition by CarlHamacher, McGraw Hill Higher Education.

Suggested reference books:

1) “Computer Architecture and Organization”, 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2) “Computer Organization and Architecture: Designing for Performance”, 10th Edition by William Stallings, Pearson Education.
3) “Computer System Design and Architecture”, 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Advance Python Programming

Course code					
Category	Core Course				
Course title	**Advance Python programming**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-II
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I

Introduction to Python Programming Language: Programming Language, History and Origin of Python Language, Features of Python, Limitations, Major Applications of Python, Getting, Installing Python, Setting up Path and Environment Variables, Running Python, First Python Program, Python Interactive Help Feature, Python differences from other languages.

Python Data Types & Input/Output: Keywords, Identifiers, Python Statement, Indentation, Documentation, Variables, Multiple Assignment, Understanding Data Type, Data Type Conversion, Python Input and Output Functions, Import command.

Operators and Expressions: Operators in Python, Expressions, Precedence, Associativity of Operators, Non Associative Operators.

Control Structures: Decision making statements, Python loops, Python control statements.

Python Native Data Types: Numbers, Lists, Tuples, Sets, Dictionary, Functions & Methods of Dictionary, Strings (in detail with their methods and operations).

UNIT II

Python Functions: Functions, Advantages of Functions, Built-in Functions, User defined functions, Anonymous functions, Pass by value Vs. Pass by Reference, Recursion, Scope and Lifetime of Variables.

Python Modules: Module definition, Need of modules, Creating a module, Importing module, Path Searching of a Module, Module Reloading, Standard Modules, Python Packages.

Exception Handling: Exceptions, Built-in exceptions, Exception handling, User defined exceptions in Python.

UNIT III

File Management in Python: Operations on files (opening, modes, attributes, encoding, closing), read() & write() methods, tell() & seek() methods, renaming & deleting files in Python, directories in Python, CSV files and Data Files.

Classes and Objects: The concept of OOPS in Python, Designing classes, Creating objects, Accessing attributes, Editing class attributes, Built-in class attributes, Garbage collection, Destroying objects.

Arrays and Matrices: The NumPy Module, Creating Arrays and Matrices, Copying, Arithmetic Operations, Cross product & Dot product, Saving and Restoring, Matrix inversion, Vectorized Functions.

UNIT IV

2D & 3D Data Visualization: The Matplotlib Module, Multiple plots, Polar plots, Pie Charts, Plotting mathematical functions, Sine function and friends, Parametric plots, Astroid, Ellipse, Spirals of Archimedes and Fermat, Polar Rose, Power Series & Fourier Series, 2D plot using colors, Fractals, Meshgrids 3D Plots, Surface Plots & Line Plots, Wire-frame Plots, Mayavi, 3D visualization.

Python and Databases: ODBC and Python, Working with Databases in MySQL, Working with Tables in MySQL, Managing users in MySQL, Accessing MySQL data from Python, Working with SQLite Database.

Suggested Readings:

1. Python for Education – Ajith Kumar B. P., Inter University Accelerator Center, New Delhi, 2010

2. Python Training Guide – Mercury Learning & Information USA, BPB Publications, 2015

3. Learn to Program, University of Toronto: https://www.coursera.org/learn/learn-to-program

4. Spoken Tutorial - IIT Bombay: https://swayam.gov.in/course/4178-spoken-tutorial-python-english 5. Python Cookbook: Recipes for Mastering Python 3, 3rd Edition - David Beazley & Brian K. Jones, O'Reilly Media, Inc., 2013

Advance Python Programming Lab

Course code					
Category	Core Course				
Course title	**Advance Python programming Lab**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-II
	0	**0**	4	**2**	
Class work	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Practical list should be prepared based on the content of the subject with following guidelines in mind.

1. Entire syllabus should be covered.

2. Practical list should be designed with real life examples.

3. List should be prepared to cover individual concepts and integration of different concepts on real life problems.

Data Warehouse and Data Mining

Course code					
Category	Core Course				
Course title	**Data Warehouse and Data Mining**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-II
	3	**1**		**4**	
Class work	25 Marks				
Exam	75 Marks				

Total	100 Marks
Duration of Exam	03 Hours

UNIT I

Data Warehouse: Need for data warehouse, Definition, Goals of data Warehouse, Challenges faced during Warehouse Construction, Advantages,

Types of Warehouse: Data Mart, Virtual Warehouse and Enterprise Warehouse. Components of Warehouse: Fact data, Dimension data, Fact table and Dimension table, Designing fact tables.

Pre-requisite Phases: Extract, Transform and load process. Warehouse Schema for multidimensional data: star, snowflake and galaxy schemas.

UNIT II

Data warehouse and OLAP technology: Difference between OLTP and OLAP, Strengths of OLAP, Applications of OLAP. Multidimensional data models: Data Cubes & Data Cuboids, Lattice.

OLAP operations: Advantages, Types: Roll up, Drill down, Pivot, Slice & Dice operations, Applications. OLAP Server: Need, Types: ROLAP, MOLAP and HOLAP, Features. Data warehouse Implementation, Introduction to Efficient computation of data cubes.

UNIT III

Data preprocessing: Need, Preprocessing stages: Data integration, Data Transformation, Data Reduction, Discretization and Concept Hierarchy Generation, Data mining primitives, Types of Data Mining, Architectures of data mining systems.

Data Characterization: Data generation & Summarization based characterization, Analytical characterization, Mining class comparisons.

Mining Association Rules in large databases: Association Rule mining, Single dimensional Boolean association rules from Transactional DBS, Multi level association rules from transaction DBS, Multidimensional association rules from relational DBS and DWS, Correlation analysis, Constraint based association mining.

UNIT IV

Classification and Prediction: Basic Classification & Prediction Model, Difference between Classification & Prediction. Classification Algorithms: Decision tree induction, Back propagation, Bayesian classification, classification based in association rules.

Prediction Algorithms: Regression approach: Linear & Non Linear regression. Classifier Accuracy & Predictor Error Measures. Cluster analysis: Purpose, Types: Partitioning and Hierarchical methods, Density based methods. Applications of Data Mining: Web mining, Temporal and Spatial data mining.

Suggested Readings:

1. W.H.Inmon: Building Data Ware House, John Wiley & Sons.

2. S . Anahory and D.Murray: Data warehousing, Pearson Education, ASIA.

3. Jiawei Han & Micheline Kamber: Data Mining - Concepts & Techniques, Harcourt India PVT Ltd. (Morgan Kaufmann Publishers).

4. Michall Corey, M.Abbey, I Azramson & Ben Taub: Oracle 8i Building Data Ware Housing, TMH.

5. I.H. Whiffen: Data Mining, Practical Machine Cearing tools & techniques with Java (Morgan Kanffmen)

6. Sima Yazdanri & Shirky S. Wong: Data Ware Housing with oracle.

7. A.K. Pujari: Data Mining Techniques, University Press.

8. IBM An Introduction to Building the Data Warehouse, PHI Publication.

9. Pieter Adriaans Dolf Zantinge: Data Mining, Addition Wesley.

10. David Hand, Heikki Mannila, and Padhraic Smyth: Principles of Data Mining, PHI Publication.

11. Anahory S., Murray D. :Data Warehousing in the Real World, Addison Wesley.

Semester 3rd

Compiler Design

Course code					
Category	Core Course				
Course title	**Software Testing and Quality Assurance**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-III
	3	**1**	0	**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT 1

Introduction to Compilers: Language Processors, The Structure of compiler: its different phases, Compiler Construction Tools, Applications of Compiler Technology.

Lexical Analysis: Role of lexical analyzer, Input Buffering, Specification and recognition of tokens, design of lexical analyzer, regular expressions, A language specifying lexical analyzer, Finite automata, conversion from regular expression to finite automata, and vice versa, minimizing number of states of DFA, Implementation of lexical analyzer.

UNIT 2

Syntax Analysis: Role of parsers, context free grammars.

Parsing Technique: Shift-reduce parsing, Operator precedence parsing, Top down parsing, Predictive parsing.

UNIT 3

LR parsers, SLR, LALR and Canonical LR parser.

Syntax Directed Translations: Syntax directed definitions, construction of syntax trees, syntax directed translation scheme, implementation of syntax directed translation, IntermediateCode Generation: three address code, quadruples and triples.

UNIT 4

Symbol Table & Error Detection and Recovery: Symbol tables: its contents and data structure for symbol tables; trees, arrays, linked lists, hash tables. Errors, lexical phase error, syntactic phase error, Semantic error.

Code Optimization & Code Generation: Code generation, forms of objects code, machine dependent code, optimization, register allocation for temporary and user defined variables.

Suggested Text Books:

1. Compilers Principle, Techniques & Tools - Alfread V. AHO, Ravi Sethi & J.D. Ullman; 1998 Addison Wesley.

Suggested Reference Books:

1. Theory and practice of compiler writing, Tremblay & Sorenson, 1985, Mc. Graw Hill.

2. System software by Dhamdere, 1986, MGH.

3. Principles of compiler Design, Narosa Publication

4. Elements compiler Design, Dr. M. Joseph, University Science Press

Machine Learning

Course code					
Category	Core Course				
Course title	**Machine Learning**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-III
	3	**1**	0	**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT I

INTRODUCTION – Learning , Machine Learning, Machine Learning Applications, History of ML, Life cycle of Machine Learning, Machine Learning and Data Science ,AI, Types of Learning, Supervised Machine Learning, Unsupervised Machine Learning, Supervised vs Unsupervised Learning, Advantages of Machine Learning, Disadvantages of Machine Learning, Install Anaconda & Python, AI vs Machine Learning, How to Get Datasets, Data Pre-processing.

UNIT 2

REGRESSION: Supervised Learning; Regression Analysis, Linear Regression, Simple Linear Regression, Multiple Linear Regression, Polynomial Regression, Underfitting and Overfitting, Advantages of Using Linear Regression, Limitations of Linear Regression, Logistic Regression,

UNIT 3

DECISION TREE LEARNING -Classification; Logistic Regression, Decision tree learning, Types of Decision Tree; Classification, Regression, Decision tree learning algorithm, Advantages of Decision tree learning, Entropy, Information gain, Issues in Decision tree learning**.**

SUPPORT VECTOR MACHINE: Introduction, Types of support vector kernel – (Linear kernel, polynomial kernel, and radial basis kernel), Hyperplane – (Decision surface), Properties of SVM, and Issues in SVM, Random forest.

UNIT 4

BAYESIAN LEARNING - Probability Fundamentals; joint probability, conditional Probability, Bayes theorem, Concept learning, Naïve Bayes classifier and its applications.

CLUSTERING ; k-means clustering, k-Nearest Neighbor Learning, Association rule learning, Apriori algorithm, Neural networks

Suggested Text Books:

1. Tom Mitchell, "Machine Learning", McGraw Hill, 1997, ISBN 0070428077
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin.

Suggested Reference Books:

1. Richard o. Duda, Peter E. Hart, and David G. Stork, "Pattern Classification", John Wiley Asia, 2006

2. T. Hastie, R. Tibshirani, & J. H. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer Verlag, 2001.

3. Ian H. Witten & Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann, 1999.

4. S. M. Weiss & C. A. Kulikowski, "Computer Systems that Learn", Morgan Kaufman Publishers, San Fancisco, CA, 1991

Advance Java Programming

<table>
<tr><td>Course code</td><td colspan="5"></td></tr>
<tr><td>Category</td><td colspan="5">Core Course</td></tr>
<tr><td>Course title</td><td colspan="5"><b>Advance Java programming</b></td></tr>
<tr><td rowspan="2"><b>Scheme and Credits</b></td><td><b>L</b></td><td><b>T</b></td><td><b>P</b></td><td><b>Credits</b></td><td rowspan="2">Semester-III</td></tr>
<tr><td><b>3</b></td><td><b>1</b></td><td>0</td><td><b>4</b></td></tr>
<tr><td>Class work</td><td colspan="5">25 Marks</td></tr>
<tr><td>Exam</td><td colspan="5">75 Marks</td></tr>
<tr><td>Total</td><td colspan="5">100 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

UNIT-I

Servlets: The life cycle of Servlet, Java Servlet Development kit, Servlet API, Reading the servlet parameters, Reading initialization parameters, Handling HTTP requests and responses, Using cookies, Session tracking and security issues.

UNIT-II

Java Server Pages (JSP): JSP Architecture, Life cycle of JSP, JSP syntax basics– Directives, Declarations, Scripting, Standard actions, Custom tag libraries, Implicit objects, Object scope. Synchronization issues, Session management.

UNIT-III

Struts: Introduction to struts framework, understanding basic architecture of Model, view, controller. Deploying the application in struts with database connectivity.

Hibernate : Introduction to hibernate framework, understanding basic architecture of Model, view, controller. Basic concepts of creating pojo files, reverse mapping, object creation in hibernate ,database connectivity .

UNIT-IV

Enterprise Java Bean: The bean developer kit (BDK), Use of JAR files, The java beans API, Creating a JavaBean, Types of beans, Stateful session bean, Stateless session bean, Entity bean.

Remote Method Invocation: Defining the remote interface, Implementing the remote interface, Compiling and executing the server and the client.

References:

1. Herbert Schildt , “The Complete Reference Java 2” , Tata McGraw -Hill.

2. H.M. Deital, P.J. Dietal and S.E. Santry, “Advanced Java 2 Platform HOW TO PROGRAM”, Prentice Hall

Advance Java Programming Lab

Course code					
Category	Core Course				
Course title	**Advance Java programming Lab**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-III
	0	**0**	4	**2**	
Class work	50 Marks				
Exam	50 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

1. Create a Servlet to handle HTTP Requests and Responses.

2. Implementation of the concept of Cookies and Session Tracking.

3. Illustrate the concept of JavaServer Pages (JSP).

4. Create a JavaBean by using Bean Developer Kit (BDK).

5. Implementation of various types of beans like Session Bean and Entity Bean.

6. Introduction to Struts platform with basic connectivity.

7. Deploying first sample program using MVC architecture in struts.

8. Implementing database connectivity in struts.

9. Creating one sample application in struts.

10. Introduction to Hibernate framework.

11. Creating simple Hibernate application.

Soft Computing

Course code					
Category	Core Course				
Course title	**Neural Networks**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-III
	3	**1**	0	**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT-I

Introduction: Introduction to soft computing, introduction to biological and artificial neural network; introduction to fuzzy sets and fuzzy logic systems.

Introduction to Genetic Algorithm: Genetic Operators and Parameters, Genetic Algorithms in Problem Solving, Theoretical Foundations of Genetic Algorithms, Implementation Issues.

UNIT-II

Artificial neural networks and applications: Different artificial neural network models; learning in artificial neural networks; neural network applications in control systems. Neural Nets and applications of Neural Network.

UNIT-III

Fuzzy systems and applications: fuzzy sets; fuzzy reasoning; fuzzy inference systems; fuzzy control; fuzzy clustering; applications of fuzzy systems. Neuro-fuzzy systems: neuro-fuzzy modeling; neuro-fuzzy control.

UNIT-IV

Applications: Pattern Recognitions, Image Processing, Biological Sequence Alignment and Drug Design,Robotics and Sensors, Information Retrieval Systems, Share Market Analysis, Natural Language Processing.

References:

1. M. Mitchell: An Introduction to Genetic Algorithms, Prentice-Hall.

2. J.S.R.Jang, C.T.Sun and E.Mizutani: Neuro-Fuzzy and Soft Computing, PHI, Pearson Education.

3. Timothy J.Ross: Fuzzy Logic with Engineering Applications, McGraw-Hill.

4. Davis E.Goldberg: Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley.

5. S. Rajasekaran and G.A.V.Pai: Neural Networks, Fuzzy Logic and Genetic Algorithms, PHI.

6. D. E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley.

Software Testing and Quality Assurance

Course code					
Category	Core Course				
Course title	**Software Testing and Quality Assurance**				
Scheme and Credits	**L**	**T**	**P**	**Credits**	Semester-III
	3	**1**	0	**4**	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

UNIT-I

Software Testing and the related concepts: significance and potential; Testability and features of Test cases. Software Testing techniques; WBT, BBT, Ticking Box testing; static analysis, symbolic testing, program mutation testing, input space , partitioning, functional program testing, data flow guided testing.

UNIT-II

Software Testing Strategies: Approach, Issues; integration, incremental, System, alpha, Beta testing etc;

Comparative evaluation of techniques: Testing tools; Dynamic analysis tools, test data generators, Debuggers, test drivers etc.. Technical Metrics for Software: Quality Factors, framework; Metrics for analysis, design, testing source code etc.

UNIT-III

Object Oriented Testing: OOT strategies and issues, Test Case design, interface testing. Software Quality Assurance: concept, importance and essence; FTR, structured walk through technique etc.

UNIT-IV

SW Reliability and SQA: SW Reliability, validation, Software Safety and Hazards Analysis; Features affecting software quality, SQA Plan. Using project management software tools, Quality management, issue, standards and methods. ISO Quality models: ISO 9000 and SEICMM and their relevance.

References:

1. Meyers, G.: The art of Software Testing, Wiley-Inter-Science.

2. Deutsch, Willis: Software Quality Engineering: A Total Technical and Management Approach, Prentice Hall.

3. Pressman : Software Engineering, TMH.

4. Gill N.S.: Software Engineering – Reliability, Testing and Quality Assurance, Khanna Book Publishing Co.(P) Ltd, N. Delhi

Semester 4th

MAJOR PROJECT REPORT

Max Marks: 500
Total Credits: 24

Course Outcomes:

At the end of the course / On completion of the course, the students will be able to:

CO1 Use of various software engineering principles used in developing programming solutions to a system.

CO2 Identify the programming technologies: languages and database etc to be used for developing a software solution.

CO3 Understand and analyze the work schedule and its phases to develop a Project.

CO4 Implement the software design in the chosen programming languages/database etc.

CO5 Test the code for validation and verification of user requirements of the software. Work in a team for software development.

Guidelines:

1. Each student should carry out Project using the software development tools /languages/ technologies that they have learnt and/or have studied during the concerned semester or any other development tools in view of the ongoing Software Industry trends.

2. It should be done by the student in an organization/college under the supervision of the staff(s) assigned by Head of the Department/Director/Principal.

3. The Project has to be assigned to the students in the beginning of the 4th Semester.