ASSAM UNIVERSITY, SILCHAR

DEPARTMENT OF COMPUTER SCIENCE

Curriculum

for

PG Programme

M. Sc. (INTEGRATED) 5 YEAR & M.Sc. (LATERAL ENTRY) 2 YEAR

Under NEP 2020

w.e.f. 2022-23

Department of Computer Science

Curriculum for PG Programme

M. Sc. (Integrated) 5-year Programme and M.Sc. 2 year (lateral entry) to the PG Semesters of the M. Sc. Integrated 5-year programme in Computer Science under NEP 2020 w.e.f. 2022-23

The Department of Computer Science offers B. Sc. Computer Science and M. Sc. Computer Science Degrees under the M. Sc. (Integrated) 5-year programme with a provision of lateral exit with a B. Sc. Degree in Computer Science after successful completion of 3 years (Six Semesters) of study and also admits students through a lateral entry into the PG level semesters of the M. Sc. (Integrated) 5-year programme leading to an M. Sc. Degree in Computer Science.

For a student to get promoted to the PG level semester of the Integrated M. Sc. Course in Computer Science, he/she must successfully complete/pass all the papers up to the 6th semester of the M. Sc. (Integrated) 5-year programme. Students who do not exercise the exit option after a B. Sc. Degree in Computer Science are awarded B. Sc. and M Sc. at the end of the successful completion of 5 years of study.

Eligibility Criteria for lateral entry into the PG level semesters of the M. Sc. (Integrated) 5-year programme:

Students who have passed four-year B. Sc. (Hons with research) degree under NEP in Computer Science with at least 50% of marks from a University recognized by UGC shall take admission in PG Second Year.

OR

Students who have passed a B.Sc. (Major) degree in computer science with at least 50% of marks from a university recognized by UGC shall take admission in PG First Year.

OR

Students who have passed three-year B. Sc. Degree under NEP in Computer Science with at least 50% of marks from a University recognized by UGC shall take admission in PG First Year.

OR

Students who have passed a B.Sc. (Pass) degree with Computer Science and Mathematics as compulsory subjects with at least 50% of marks from a University recognized by UGC shall take admission in PG First Year.

OR

Students who have passed a Bachelor in Computer Applications (BCA) degree with at least 50% of marks from a University recognized by UGC and also have passed the Higher Secondary/ (10+2) level examination in Science Stream with Mathematics and Physics as compulsory subjects shall take admission in PG First Year.

OR

Students who have passed a Bachelor in Information Technology with at least 50% of marks from a University recognized by UGC and also have passed the Higher Secondary/(10+2) level examination in Science Stream with Mathematics and Physics as compulsory subjects shall take admission in PG First Year.

DEPARTMENT OF COMPUTER SCIENCE
Curriculum for PG Programme

M. Sc. (INTEGRATED) 5 YEAR & M.Sc. (LATERAL ENTRY) 2 YEAR

Under NEP 2020 w.e.f. 2022-23

SEMESTER I

Sl. No.	Paper Code	Subject	Credit	L T P	Sessional Marks	End Semester Marks	Total Marks
1.	MCS-500	Orientation	Nil	- - -	-	-	-
2.	MCS-501 (Core)	Principles of Compiler Design	4	4 - -	30	70	100
3.	MCS-502 (Core)	Design and Analysis of Computer Algorithms	4	4 - -	30	70	100
4.	MCS-503 (Core)	Cryptography	4	4 - -	30	70	100
5.	MCS-504 (SEC)	Fuzzy Set Theory and Applications/Online (Swayam/UGC approved MOOC)	3	3 - -	30	70	100
6.	MCS-505 (ALIF)	a) Lab on Principles of Compiler Design b) Lab on Design and Analysis of Computer Algorithms	3	- - 6	15 15	35 35	50 50
7.	MCS-506 (CCEC)	Compulsory Community Engagement	2	1 - 2	30	70	100
Total			**20**	**24** (Hours/ week)	**165**	**385**	**600**

SEMESTER- II

Sl. No.	Paper Code	Subject	Credit	L T P	Sessional Marks	End Semester Marks	Total Marks
1.	MCS-551 (Core)	Theory of Computation	4	4 - -	30	70	100
2.	MCS-552 (Core)	Python Programming	4	4 - -	30	70	100
3.	MCS-553 (Core)	Artificial Intelligence	4	4 - -	30	70	100
4.	MCS-554 (IDC)	Internet Technology/Online (Swayam/UGC approved MOOCs)	3	3 - -	30	70	100
5.	MCS-555 (ALIF)	a) Lab on Python Programming b) Lab on Artificial Intelligence	3	- - 6	15 15	35 35	50 50
6.	MCS-556 (VBC)	Cyber Security/Online (Swayam/UGC approved MOOCs)	2	2 - -	30	70	100
Total			**20**	**23** (Hours/ week)	**165**	**385**	**600**
SEMESTER – I+II Total Marks: 1100					**Post-Graduate Diploma: 40 Credits**		

SEMESTER- III

Sl. No.	Paper Code	Subject	Credit	L T P	Sessional Marks	End Semester Marks	Total Marks
1.	MCS-601 (Core)	Data Mining and Knowledge Discovery	4	4 - -	30	70	100
2.	MCS-602 (IDC)	Programming in Java	4	4 - -	30	70	100
3.	MCS-603 (ECC)	(A)Wireless & Mobile Computing/(B) Mobile & AdHoc Networks/(C) Natural Language processing/ (D)Software Engineering/(E)Distributed Computing/(F)Pattern Recognition/(G)Digital Image Processing/(F) Operation Research & Optimization (Online (Swayam/UGC approved MOOC))	4	4 - -	30	70	100
4.	MCS-604 (ALIF)	) Lab on Data Mining) Lab on Java Programming	4	- - 8	15 15	35 35	50 50
5.	MCS-605	Dissertation (Research Project Part I)	4	- - 8	30	70	100
Total			**20**	**28** (Hours/ week)	**150**	**350**	**500**

SEMESTER- IV

Sl. No.	Paper Code	Subject	Credit	L T P	Sessional Marks	End Semester Marks	Total Marks
1.	MCS-651 (Core)	Data Science	4	4 - -	30	70	100
2.	MCS-652 (Core)	Machine Learning	4	4 - -	30	70	100
3.	MCS-653 (ECC)	(A)Artificial Neural Networks/(B) Advanced Computer Architecture and Parallel Computing/(C)Quantum Computing/(D)Biologically Inspired Computing/Online (Swayam/UGC approved MOOC)	4	4 - -	30	70	100
4.	MCS-654	Dissertation (Research Project Part II)	8	- - 16	60	140	200
Total			**20**	**28** (Hours/ week)	**150**	**350**	**500**
SEMESTER –I+II+III+ IV Total Marks: 2100							**Master's Degree: 80 Credits**

Semester VII (M. Sc. Integrated 5-year programme)
and
Semester I (M.Sc. Lateral Entry 2 year)

Course Code: MCS-500 **Course Title: Orientation**

Credits: Nil **LTP: 0 – 0 – 0**

Course Objectives

1. To make aware the students about the academic and research environment of the University, along with the departmental facilities, resources and activities.
2. To guide the students for a purposeful learning during their complete tenure of programme.
3. To make the students aware about their curriculum, evaluation process, academic flexibility and extracurricular activities
4. To enlighten the students about quality of education, feedback mechanisms, grievance redressal, discipline and their rights and duties.

Course Outcomes

The course will enable the students to be aware of their overall learning procedure, resources, and methods for their optimum use.

Evaluation:

This will be a non-credit course, with contact duration spread over minimum one week, and maximum of two weeks. The student must attend 70% of the classes to pass the course, and there shall be No test or Exam. Classes for orientation courses will run along with the normal classes.

Mode of teaching:

Interaction with experts/faculty members, discussions through physical/online mode, exchange of materials - digital lectures, documents, guidelines.

Topics for interaction:

1. About the University and Department, mutual expectations of students and department to achieve Mission of Department / University. Student behaviour and discipline, anti-ragging campus, Placements
2. Curricular aspects and evaluation process, Online registration processes, Scholarships and Awards, safety / lab-safety/ biosafety at workplace.

3. Feedback mechanisms, Quality framework and role of students, Mentor and mentee system, Green campus initiatives, Waste management and Zero-plastic use in the campus.
4. Library activities, Extracurricular activities, University facilities for students and its maintenance, Student Union, Rights and Duties of the students.

5. Grievance redressal system, Gender issues and sensitization, Ethics in education, Protocols for emergencies like natural disasters and fire at workplace.

Course Code: MCS-501 **Course Title: Principles of Compiler Design**

Credits: 4 **LTP: 4 – 0 – 0**

Course Objectives

1. A brief introduction to compiler basics, including lexical analysis and syntax trees.
2. To introduce the Top-Down and Bottom-Up parsing algorithms.
3. An introduction to the symbol table and run-time storage management.
4. Describe how Intermediate Code Generation works.
5. An explanation of code generation and code optimization.

Course Outcomes

1. Understanding lexical analysis and syntax trees is essential for understanding compiler basics.
2. Understanding top-down and bottom-up parsing algorithms.
3. An understanding of the symbol table and how run-time storage is handled.
4. Getting a better understanding of intermediate code generation.
5. Understanding the generation and optimization of code.

UNIT-I: Overview of compiling process, some typical compiler structures, Regular expressions, The role of lexical analyzer, Input Buffering, Specification of tokens, recognition of tokens, Syntax trees, ambiguity, Context free Grammar and derivation of parse trees

UNIT-II: Top-down and bottom-up parsing, operator precedence parsing, syntax-directed translation, translation schemes

UNIT-III: Symbol Table: The contents of a symbol table, Data structures for symbol tables (ST), design of an ST, ST for block-structured languages
Run time storage administration: Storage allocation strategies, static, dynamic and heap memory allocation, memory allocation in block structured languages, memory allocation in recursion

Unit-IV: Intermediate Code Generation: Intermediate Languages, Intermediate Representation Techniques, Statements in three-address code, Implementation of three-address instruction, Three-address code generation, Code Generation for arrays, Translation of Boolean expressions, Translation of Control flow statements, Translation of Case Statements, Function Calls

Unit-V: Code Generation: Basic block, Representation of Basic blocks, Problems in Code Generation, Code Generation for DAGs, Register Allocation and Register Interference Graph, Code Generation using Dynamic Programming.

Code Optimization: Principal sources of optimization, loop optimization, Global data flow analysis.

Text Books/References:

1. Compilers - Principles, Techniques and tools, Alfred V Aho, R.Sethi, D. Ullman (Pearson)
2. Compiler Design - K. Muneeswaran, Oxford University Press
3. The theory and practice of compiler writing, Trembley and Sorenson, (McGraw Hill)
4. Compiler Design, Santanu Chattopadhyay, PHI

Course Code: MCS-502 Course Title: Design and Analysis of Computer Algorithms

Credits: 4 LTP: 4 – 0 – 0

Course Objectives

Upon completion of this course, students should be able to do the following:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.

Course Outcomes

Students who complete the course will be able to do the following:

1. Argue the correctness of algorithms using inductive proofs and invariants. Analyze worst-case running times of algorithms using asymptotic analysis.
2. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Solve problems that employ this paradigm.
3. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Solve problems that employ this paradigm.
4. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Solve problems that employ this paradigm.
5. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Solve problems that employ graph computations as key components, and analyze them.
6. Explain NP-completeness, approximation algorithms and the benefit of using approximation algorithms.

UNIT-I: Definition of Algorithm, asymptotic notation: standard notations and common functions, Solution of recurrences: Substitution method, recursion tree method, master method, Stacks and queues, priority queues, heap and heap sort

UNIT-II: Divide and Conquer: the general method, binary search, finding the maximum and minimum, merge sort, quick sort, strassen's matrix multiplication, Insertion and deletion in trees

UNIT-III: Graph Algorithms: Representation of Graphs, breadth first and depth first search, strongly connected components, topological sort, algorithms of Kruskal and Prim, Bipartite Graphs, Maximum matching in Bipartite Graph, String matching algorithms, String matching with finite automata

UNIT-IV: Dynamic Programming: the general method, multistage graphs, optimal binary search trees, Warshall and Floyd Algorithms, Greedy method-Knapsack problem, Huffman Trees. Iterative Improvement: the stable matching problem

UNIT-V: Computational Geometry algorithms: Line segment properties, finding the convex hull, finding the closest pair of points, NP hard and NP Complete problems

Text Books/References:

1. Introduction to Algorithms, Thomas H Cormen et al., PHI
2. Fundamentals of Computer Algorithms, Ellis Horowitz, SartajSahni, Rajasekaran, Universities Press
3. Introduction to Design and analysis of Algorithms, AnanyLevitin , Pearson
4. Algorithm Design, Jon Kleinberg, Eva Tardos, Pearson
5. Fundamentals of Algorithmics, Brassard and Bratley, PHI

Course Code: MCS-503 **Course Title: Cryptography**

Credits: 4 **LTP: 4 – 0 – 0**

Course Objectives

1. To introduce the concepts of Cryptography and Security, substitution Techniques, Permutation, techniques.
2. To explain the concepts of Encryption and Decryption techniques for symmetric and asymmetric key, Theory of Blocks, DES and its variations, and strength of DES.
3. To explain the Modern Asymmetric key Algorithm: IDEA, Blowfish, TWO-FISH, RC-5, and AES.
4. To explain the public key cryptography, prime number, RSA algorithm, Diffie-Hellman key exchange algorithm and man-in-middle attack, hashes and message digest. Message authentication: MD5, SHA1, HMAC.
5. To explain the concept of system Authentication: Kerberos v_4 and v_5, network security: Firewall, IP security, virtual private network.

Course Outcomes

After successful completion of the course, the students will be able to

1. Learn the concepts of the concepts of Cryptography and Security, substitution Techniques, Permutation, techniques.
2. Learn the concepts of Encryption and Decryption techniques for symmetric and asymmetric key, Theory of Blocks, DES and its variations, and strength of DES.

3. Learn the concept of Modern Asymmetric key Algorithm: IDEA, Blowfish, TWO-FISH, RC-5, and AES.
4. Learn the principle and working of public key cryptography, prime number, RSA algorithm, Diffie-Hellman key exchange algorithm and man-in-middle attack, hashes and message digest. Message authentication: MD5, SHA1, HMAC.
5. Learn the concepts of system Authentication: Kerberos v_4 and v_5, network security: Firewall, IP security, virtual private network.

UNIT-I: Foundations of Cryptography and Security: Approaches, Policies, Principles Ciphers and Secret Messages, Security Attacks and Services
Mathematical Tools for Cryptography: Substitution Techniques (Caesar Ciphers, Modified Caesar Ciphers, Monoalphabetic Ciphers, Monohomophonic cipher, Polygram cipher) and Permutation Techniques (Rail Fence techniques, Simple Columnar Techniques)

UNIT-II: Encryption and Decryption: Basics and Techniques (Symmetric and Asymmetric key), Steganography
Conventional Symmetric Encryption Algorithms: Theory of Block Cipher Design, Feistel Cipher Network Structures, DES and Triple DES, Modes of Operation (ECB, CBC, OFB, CFB), Strength (or Not) of DES

UNIT-III: Modern Symmetric Encryption Algorithms: International Data Encryption Algorithm (IDEA), CAST, Blowfish, Twofish, RC2, RC5, Rijndael (Advanced Encryption Standard), Key Distribution. Design of Stream Cipher, One Time Pad

UNIT-IV: Public Key Cryptography: Prime Numbers and Testing for Primality, Factoring Large Numbers, RSA, Diffie-Hellman, ElGamal, Key Exchange Algorithms, Public-Key Cryptography Standards, Hashes and Message Digests: Message Authentication, MD5, SHA, RIPEMD, HMAC

UNIT-V: Digital Signatures, Certificates, User Authentication: Digital Signature Standard (DSS and DSA)
Authentication of Systems: Kerberos V4 and V5, X.509 Authentication Service
Network Security: Firewalls, IP security, Virtual Private Networks (VPN)

Textbooks:

1. Atul Kahate : Cryptography and Network Security
2. William Stallings, Cryptography and Network Security: Principles and Practice, ISBN 0131873164, 4/e
3. Bruce Schneier, Applied Cryptography (ISBN 0471128457), 2/e
4. Alfred J. Menezes, Handbook of Applied Cryptography
5. Michael Welschenbach, Cryptography in C and C++ (ISBN 1590595025), 2/e
6. Douglas R. Stinson, Chapman & Hall, Cryptography: Theory and Practice, Third Edition CRC 2005, ISBN: 1584885084

References Books:

1. William Stallings, Cryptography and Network Security, 4th.Ed, Prentice Hall PTR, Upper Saddle River, NJ, 2006
2. Wenbo Mao, Modern Cryptography: Theory and Practice, Prentice Hall, 2004
3. Richard A. Mollin, An Introduction to Cryptography, Chapman and Hall/CRC, 2001.
4. B. Schneier, Applied Cryptography, John Wiley and Sons, NY, 1996.
5. A. Menezes, P. Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1997.
6. Thomas H. Barr, Invitation to Cryptography, Prentice Hall, 2002.
7. Richard J. Spillman, Classical and Contemporary Cryptology, Prentice Hall, 2005.

Course Code: MCS-504 **Course Title: Fuzzy Set Theory and Applications**

Credits: 3 **LTP: 3 – 0 – 0**

Course Objectives

1. To introduce the basic concepts of Fuzzy set, Fuzzy logic, and Fuzzy measure theory.
2. To introduce uncertainty problems in real-world applications.
3. To introduce the role of Fuzzy sets and Fuzzy logic in representing uncertainty and in providing the foundation for handling uncertainty, and reasoning with them in real-world applications.

Course Outcomes

Upon successful completion of this course, the student shall be able to:
1. Understand basic concepts of fuzzy sets, Fuzzy logic, and Fuzzy measure theory.
2. Handle the problems having uncertain and imprecise data.
3. Apply the concepts of Fuzzy sets to various applications viz. Fuzzy Expert System, Fuzzy Controllers, Fuzzy System & Neural Network, Fuzzy Automata, Pattern Recognition and Fuzzy Image Processing

Unit I: Introduction to Fuzzy Sets
Basic Concepts: Fuzzy Sets versus Crisp Sets, Properties of Alpha-Cuts, Representation of Fuzzy Sets

Unit II: Operation on Fuzzy Sets
Types of Operations, Fuzzy Complements, Fuzzy Intersection: t-Norms, Fuzzy Unions: t-Conorms, Combinations of Operations
Fuzzy Arithmetic: Fuzzy Numbers, Linguistic Variables, Arithmetic Operations on Intervals, Arithmetic Operations on Fuzzy Numbers

Unit III: Fuzzy Relations
Crisp *vs* Fuzzy Relations, Binary Fuzzy Relations, Binary Relation on a Single Set, Fuzzy Equivalence Relations, Fuzzy Compatibility Relations, Fuzzy Ordering Relations, Fuzzy Morphisms

Unit IV: Possibility Theory
Fuzzy Measures, Evidence Theory, Possibility Theory, Fuzzy Sets and Possibility Theory, Possibility Theory versus Probability Theory
Fuzzy Logic: Multivalued Logics, Fuzzy Propositions, Fuzzy Quantifiers, Linguistics Hedges

Unit V: Some Applications of Fuzzy Sets
Approximate Reasoning (Overview): Fuzzy Expert System, Fuzzy Implications
Fuzzy Systems (Overview): Fuzzy Controllers, Fuzzy System & Neural Network, Fuzzy Automata
Pattern Recognition (Overview): Fuzzy Clustering, Fuzzy Pattern Recognition, Fuzzy Image Processing

Text Books/References:

1. Fuzzy Sets and Fuzzy Logic Theory and Applications – George J. Klir and Bo Yuan, PHI Pvt. Ltd.
2. Fuzzy Set Theory and Its Applications – Zimmermann, Hans-Jürgen, Springer Netherlands
3. Neutral Networks in Computer Intelligence – LiMin Fu, McGraw Hill Edition
4. Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence – Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, Pearson
5. Fuzzy Logic with Engineering Applications, Timothy J. Ross, 3rd Edition, John Wiley & Sons, Ltd

Course Code: MCS-505 (a) **Course Title: Lab on Principles of Compiler Design**

Credits: 3 **LTP: 0 – 0 – 6**

Problems related to Compiler Design should be solved by using the Programming languages C/C++/JAVA as well as various tools for Compiler Construction and Design like LEX, YACC, BYSON etc. Following are some areas of Compiler Design for laboratory programming assignments but the assignments should not be limited to these only:

1. Construction of a lexical analyzer and LL(1) parser for a subset of FORTRAN/PASCAL/C/C++ (to be done without using any generator).
2. Construction of a lexical analyzer and LALR(1)/LR(1) parser for a subset of C/C++ (generators like LEX, YACC, BISON to be used).
3. A construction of a translator from a high level to an intermediate language which is also a very simple subset of C (The correctness of this translation may be checked by compiling this intermediate program by a standard compiler).
4. Construction of a target code generator from the above intermediate language program to the assembly language of a suitable target machine (e. g. Intel 8088). Addition of rudimentary code optimization (like peep-hole)/jump optimization.
5. Register optimization to the generated compiler.
6. Experiments with incorporation of debugging features.

Course Code: MCS-505 (b) Course Title: Lab on Design and Analysis of Computer

Problems related to Design and Analysis of Computer Algorithms should be solved by using the Programming languages C/C++/JAVA (preferably on Unix/Linux/Solaris operating systems environment on a network). Following are some areas of Design and Analysis of Computer Algorithms for laboratory programming assignments but the assignments should not be limited to these only:

1. Stack and queues, tree, heap and heap sort, graphs and hashing.
2. Divide and conquer method: binary search, merge sort, quick sort, matrix multiplication, minimum spanning tree.
3. Dynamic programming: multistage graphs, all pair shortest paths, optimal binary search trees/0/1 knapsack, traveling salesperson problem, flow shop scheduling.
4. Search and traversal techniques: AND/OR graphs, game trees, bi connected components and depth search.
5. Backtracking: Hamilton cycles, the fast Fourier transform, NP-HARD and NP complete problems.

Course Code: 506
Course Title: Compulsory Community Engagement

Credits: 2 LTP: 1 – 0 – 2

This is two credit field project course based on activities involving community engagements in rural areas/villages or urban areas. The field project may have computational implementation under the guidance of faculty supervisor. In this course each student/group is required to visit communities in rural areas/villages or urban areas to survey and collect data related to the livelihood, cultivation, practices, culture, rural economy, rural technology and various other aspects of rural/urban lives. Students are also required to gather information about various Government run projects and assistance programs for rural people and their performances. These include PMAY, MGNREGA, Midday meal program, Beti Bachao Beti Padhao and other such programs. In this field project work the students are expected to evaluate the success and performances of such projects run by the Government through collection and computation of data obtained by interacting and surveying people in the villages. The student/group may undertake ICT awareness progrmme, awareness on ICT enabled governments schemes, etc. Each student/group will work under a faculty supervisor/guide.

For CCEC (Compulsory Community Engagement Course), the readings from e-content and reflections from field visits should be maintained by each student in a Field Diary. Before going to the CCEC programme each student/group of students shall submit a brief proposal for CCEC activity to the mentor mentioning the plan of work to be undertaken in the community. This will be a component of his/her periodic assessment by the mentor. In line with the UGC model curriculum on CCEC, the performance of students in CCEC can be assessed as follows, (i) Participation in Field Visits will be allocated 30% marks awarded through assessment by the Mentor/Supervisor/Teacher-in-charge; (ii) field project in the form of a Report will have 40% of total marks, and (iii) presentation of field project findings will have 30% of total marks.

Evaluation process: 30% of marks will be awarded from the periodic internal assessment by the Mentor/Supervisor/Teacher-in-charge. Rest 70% of marks will be from the evaluation of the report submitted by the student (with viva voice or presentation). This evaluation will be done by the DAC or a Board of Examiners constituted by the Department, with at least one examiner from allied discipline in either case.

Semester II

Course Code: MCS-551 **Course Title: Theory of Computation**

CREDIT: 4 **L-T-P: 4-0-0**

Course Objectives
This course focuses on the basic theory of Computer Science and formal methods of computation like automata theory, formal languages, grammars and Turing Machines. The objective of this course is to explore the theoretical foundations of computer science from the
perspective of formal languages and classify machines by their power to recognize languages.

Course Outcomes
Students who complete the course will be able to do the following:

Understand the basic properties of formal languages and grammars.
Differentiate regular, context-free and recursively enumerable languages.
Understand and implement grammars to produce strings from a specific language.
Understand concepts relating to Turing machines, decidability and intractability.

UNIT-I: Sets, Relations and Functions, Fundamental Proof Techniques, Alphabets, Strings and languages, Finite and Infinite sets, Finite Representation of Languages, Regular Expressions, Deterministic and Nondeterministic Finite Automata (DFA and NFA), Equivalence of DFA and NFA.

UNIT-II: Properties of the languages Accepted by Finite Automata, State Minimization of a DFA, Pumping Lemma for Regular Sets, Regular and Non-regular languages.

UNIT-III: Context-free Grammars, Parse Trees, Regular Language and context free language, Chomsky's Normal Form, Pushdown Automata, Properties of Context Free Languages, Pumping Lemma for Context Free Languages, Determinism and Parsing.

UNIT-IV: The definition of a Turing Machine, Computing with TM, Recursive and Recursively Enumerable Language, Extensions of Turing Machines, Non Deterministic Turing Machines, Chomsky's Hierarchy.

UNIT-V: Primitive and n-Recursive Function, Church's thesis, The Halting problem, Unsolvability, Computational Complexity.

Text Books/References:

1. H.R.Lewis & C.H. Papadimitriou: Elements of The Theory of Computation, P.H.I.
2. J.E.Hopcroft, R.Motwani & J.D.Ullman : Introduction To Automata Theory, Language and Computation ,Pearson Education.
3. K.L.P.Mishra, N.Chandrasekaran: Theory of Computer Science (Automata, Languages And Computation), PHI.
4. John Martin: Introduction to languages and Theory of Computation, McGraw Hill
5. D.A.Cohen : Introduction To Computer Theory (J. Wiley)

Course Code: MCS-552 **Course Title: Python Programming**

CREDIT: 4 **L-T-P: 4-0-0**

Course Objectives

To understand why Python is a useful scripting language for developers; to learn how to use lists, tuples, and dictionaries in Python programs; to learn how to use indexing and slicing to access data in Python programs ; to learn how to write loops and decision statements in Python; to learn how to write functions and pass arguments in Python; to learn how to build and package Python modules for reusability; to learn how to read and write files in Python; to learn how to design object-oriented programs with Python classes ; to learn how to use class inheritance in Python for reusability; to learn how to use exception handling in Python applications for error handling ; to learn how to use Machine Learning tools like Pytorch, keras, TensorFlow etc.

Course Outcomes

At the end of the course it is expected that a student would be reasonably proficient in writing Python programs for solving various problems as the course covers topics ranging from basics of Python Programming to advanced level.

UNIT-I

Introduction: Basic Elements of Python, Operators, Python Namespace, Python Statements & Comments, Python Type Conversion, Indentation in Python, Python IDEs

Python Flow Control: if…else, for loop, while loop, break and continue, Python Pass, range statement

UNIT-II

Python Functions: Python Functions, Function Arguments, Recursion, Anonymous Function, Lambda function, Global, Local and Nonlocal, Python Global Keyword, Python Modules, Python Package.

Python Files: Python File Operation, Python Directory, Python Exception Handling, User defined exception. Assertions

UNIT-III

Python Collections: List, Tuple, Sets and Dictionary; String Manipulation: Basic Operations, Slicing, Python Regular expressions; Python iterators, Python Generators, Python Closure, Python Decorators, List Comprehension

UNIT-IV

Python Object & Class Inheritance, Python Operator Overloading, Instance Methods, Exceptions; MATPLOTLIB; TKINTER

UNIT-V

Python packages: NUMPY, PANDAS, SCIKIT-LEARN, PyTorch, KERAS, TENSOR FLOW

TEXT BOOKS

1. Introduction to Computation and Programming Using Python, John V. Guttag, PHI
2. Core Python Programming, Dr. R.Nageswara Rao, dreamtech Press.

Course Code: MCS-553 **Course Title: Artificial Intelligence**

Credits: 4 **LTP: 4 – 0 – 0**

Course Objectives

1. To introduce the historical evolution of Artificial Intelligence (AI), its foundations and applications.
2. To introduce the basic principles, techniques, and applications of AI and machine Learning (ML).
3. Experiment implementation with an AI programming language (Prolog/Python).
4. Explore the scope, potential, limitations, and implications of intelligent systems.

Course Outcomes

Upon successful completion of this course, the student shall be able to:

1. Demonstrate fundamental understanding of the history of Artificial Intelligence (AI), its foundations and applications.

2. Apply basic principles of AI techniques in solutions that require problem solving, inference, perception, knowledge representation, and learning.
3. Formalize and design solutions to practical problems using the AI and ML strategies introduced during the course.
4. Demonstrate proficiency developing applications in an AI programming language (Prolog/Python).

UNIT-I: Introduction

Foundations of Artificial Intelligence (AI), History of AI, State of the Art, Risks and Benefits of AI, Intelligent Agents, Brief discussion of applications of AI (Expert System, Natural Language Processing, Speech and Pattern Recognition etc.), Problems, Problem Spaces, and Search: Defining the Problem as a State Space Search, Production Systems, Control Strategies, Problem Characteristics, Issues in the Designing of Search Programs

UNIT-II: Problem-solving

Solving Problems by Searching - Problem-Solving Agents, Example Problems, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Search in Complex Environments, Adversarial Search and Games, Constraint Satisfaction Problems

UNIT-III: Knowledge, reasoning, and planning

Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation, Automated Planning, Uncertain knowledge and reasoning - Quantifying Uncertainty, Probabilistic Reasoning

UNIT-IV: Machine Learning

Forms of learning - Supervised Learning, Unsupervised Learning, Reinforcement Learning; Learning Probabilistic Models, Deep Learning

UNIT-V: Application areas and AI languages

Computer Vision, Natural Language Processing, Robotics, etc.; Philosophy, Ethics, and Safety of AI

AI languages and their important characteristics - Prolog/Python

Text Books/References:

1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 4th Edition, Pearson Education
2. Artificial Intelligence, E. Rich, K. Knight, S. B. Nair, 3rd Edition, McGraw Hill Education
3. Artificial Intelligence, P. H. Winston, 3rd Edition, Pearson
4. Introduction to Artificial Intelligence, E. Charniak & D. McDermott, A. Wesley
5. PROLOG Programming Techniques and Application, S. Garavaglia, Harper and Row
6. Introduction to Turbo PROLOG, Carl Townsend, Sybex Inc., U.S.
7. PROLOG: Programming for Artificial Intelligence, BRATKO, 3rd Edition,Pearson Education India

8. The Complete reference Python, Martin C. Brown, 4th Edition, McGraw-Hill
9. Problem Solving and Python Programming, E. Balagurusamy, McGraw-Hill
10. Core Python Programming, Dr. R. Nageswara Rao, 3rd Edition, Dreamtech

Course Code: MCS-554 **Course Title: Internet Technology**

Credit: 3 **LTP: 3-0-0**

Course Objectives:

1. Introduction of Internet and basic components of Internet.
2. To explain the fundamentals of HTML and HTML 5.
3. To explain the basic properties of CSS and CSS3 in html file
4. To explain the basic properties of JavaScript in html file
5. To explain dynamic web page using JavaScript

Course Outcomes:

Upon successful completion of the course, the students will:

1. Get familiar with the basics of Web Designing & Publishing.
2. Gain deep understanding of the implementation of HTML tags and be able to create HTML static web pages.
3. Able to use basic properties of JavaScript in html file
4. Able to create dynamic web page using JavaScript

Unit I: Basics of Web Design

Overview of Internet and WWW, Basic elements of the Internet, Internet services, Internet Browsers and Servers, Introduction to WWW, URL, webpage, web site, web servers, web browser, Web Application, Client and Server-side scripting languages, types of websites, Web Design and Development, Internet Addressing: standard Internet Address, Domain Name Server (DNS).

Unit II: Web Page Development using HTML5

HTML Fundamentals: HTML & its relevant history, Anatomy of an HTML Tag, Basic HTML Document Structure, working with Text, working with Lists, Tables and Frames, working with Hyperlinks, Images and Multimedia, Working with Forms and Controls. Advanced Elements in HTML5: Semantic Elements, New Input Type Elements, Multimedia Tags

Unit III: Introduction to Cascading Style Sheets

Introduction to Cascading Style Sheet (CSS), basic syntax and structure, CSS selectors, Ways of specifying style, CSS Properties, CSS Styling (Background, Text, Fonts, Lists, Tables, Links), CSS Box Model, CSS Navigation Bar; CSS3: - CSS Rounded Corners, Box & Text Shadow, Gradients, Background Images, Transitions, Transforms, and Animations, CSS Layout

Unit IV: JavaScript Programming I

Introduction to Client-Side Scripting, Basics of Java Script, Java Script Statements, Comments, variables, Operators and Expressions, Conditions statement, Functions, Dialog boxes.

Unit V: JavaScript Programming II

The Java Document Object Model (DOM): JavaScript Document Object Model hierarchy – Create, find and manipulate HTML Element using Objects and methods. Form validation, Applying Style using JavaScript. Creating New window, Accessing & manipulating, History of HTML Pages; Forms: Form object, built in objects, User defined objects, Cookies, Java Script Window; DHTML: Introduction to DHTML, DHTML CSS, DHTML Java Script, DHTML HTML DOM, DHTML Events.

Text Books:

1. Matthew MacDonald: HTML5 The Missing Manual, O'Reilly Media, August 2011.
2. Peter Gasston: The Book of CSS3, A Developer's Guide to the Future of Web Design, No Starch Press, April 2011.
3. Richard York: Beginning CSS Cascading Style sheets for Web Design, Wrox Press (Wiley Publishing), 2005.

Reference Books:

1. Ivan Bayros: Web Enabled Commercial Application Development using HTML, JavaScript, DHTML and PHP, fourth revised edition, BPB Publication.
2. David Mc Farland: CSS The Missing Manual, O'Reilly, 2006.
3. Julie C. Meloni: HTML, CSS and JavaScript All in One, Pearson.

Web References:

1. http://www.tutorialspoint.com/html5 [For notes on HTML5 tags]
2. http://www.w3schools.com/html/ [For HTML5, CSS and JavaScript notes and examples]
3. https://in.godaddy.com/help/dreamweaver-cs6-publish-your-website-7811[Publish your website using Dreamweaver]
4. http://fullbooksfreedownload.blogspot.in/2016/02/html-css-javascript-web-publishing-in.html [Book :- HTML, CSS & JavaScript Web Publishing in One Hour a Day, Sams Teach Yourself, 7th Edition PDF]

Course Code: MCS-555 **Course Title: (a) Lab on Python Programming**

CREDIT: 3 **L-T-P: 0-0-6**

1. Problems related to if….else structure of Python, problems related to looping, break and continue, problems to identify the usage of pass and range statements in python.
2. Problems related to usage of functions in Python, Global, local and Non local functions.
3. Problems related to Recursion, Anonymous Function, Lambda function, Python Modules and Python Package.
4. Problems related to File handling in Python, exception handling, usage of User defined exception and Assertions
5. Problems related to Python Collections: List, Tuple, Sets and Dictionary; Problems related to String Manipulation, Basic Operations, Slicing, Python Regular expressions; Python iterators, Python Generators, Python Closure, Python Decorators, List Comprehension
6. Problems related to Python Object & Class Inheritance, Python Operator Overloading, Instance Methods, Exceptions; Problems on MATPLOTLIB and TKINTER
7. Problems on various Python packages /Machine Learning Tools: NUMPY, PANDAS, SCIKIT-LEARN, PyTorch, KERAS, TENSOR FLOW

Course Code: MCS-555 **Course Title: (b) Lab on Artificial Intelligence**

Lab on Artificial Intelligence Problems should be solved by using the programming language Prolog/Python

1. Write a Program to Implement Breadth First Search.
2. Write a Program to Implement Depth First Search
3. Write a Program for Pre-order, In-order and Post-order traversal of binary trees
4. Write a program to implement Hill Climbing Algorithm
5. Write a program to implement A* Algorithm
6. Write a Program to Implement Tic-Tac-Toe game
7. Write a Program to Implement 8-Puzzle problem
8. Write a Program to Implement Tower of Hanoi
9. Write a Program to Implement Water-Jug problem
10. Write a Program to check whether a given list is palindrome or not?
11. Write a Program to Implement Travelling Salesman Problem
12. Write a Program to Implement Monkey Banana Problem
13. Write a Program to Implement Missionaries-Cannibals Problems
14. Write a Program to Implement 4-Queens Problem
15. Write a Program for Quick Sort using cut in Prolog.
16. Given the following facts:
    Steve only likes easy course.
    Science course are hard.
    All the courses in the basket weaving department are easy.
    BK301 is a basket weaving course.

Write a Program to find “What course would Steve like?”

17. Given the following facts:

Sam likes all Indian mild food.
Sam likes all Chinese food.
Sam likes all Italian food.
Sam likes chips.
Curry, dal, tandoori, kurma are Indian food.
Dal, tandoori, kurma are Indian mild food.
chowmein, chopsuey, sweetandsour are Chinese food.
pizza and spaghetti are Italian food.

Write a Program to find a) What foods does Sam like?, b) Does Sam like Curry?, c) Does Sam like Chips?

18. Given the following facts:

All people who are not poor and are smart are happy.
Those people who read are not stupid but smart.
John could read and is wealthy.
Happy people have exciting lives.

Write a Program to prove “John has an exciting life”.

19. Given the following facts:

John likes all food.
Apples are food.
Chicken is food.
Anything anyone eats & isn’t killed by is food.
Bill eats Peanuts and is still alive.
Sue eats everything Bill eats.

Write a Program to prove that “John likes peanuts” and find “what food does Sue eat?”

20. Given the following facts:

Marcus was a man.
Marcus was a Pompeian.
All Pompeian’s were Roman.
Caesar was a ruler.
All Roman were either loyal to Caesar or hated him.
Everyone is loyal to someone.
People only try to assassinate ruler they are not loyal to.
Marcus try to assassinate Caesar.
All men are people.

WAP to find "Is Marcus loyal to Caesar?", "Does Marcus hate Caesar?"

Course Code: MCS-556 **Course Title: Cyber Security**

CREDIT: 2 **L-T-P: 2-0-0**

Course Objectives:
This programme aims to provide a foundational platform for Cyber Security Aspirants by providing Cyber Security Awareness and minimizing damage to the resources and ensuring the protection of information technology assets. The programme focuses on introduction to Concepts of Security and Internet Security protocols.

Course Outcomes:
Upon successful completion of the programme, candidates will be familiar with cyber security landscapes and able to
a) Analyze and evaluate the cyber security needs of an organization.
b) Learn types of attack, Need for security, Security Approaches, Principal of security.
c) Learn Internet security protocols such as Secure Socket Layer (SSL), Transport Layer Security (TLS), Secure hyper text transfer protocol, Secure Electronics Transaction (SET).
d) Learn the security provided by GSM, 3G, 4G, and 5G along with Indian IT act 2000 and the security initiatives in India.

UNIT-I
Introduction to The Concepts of Security: The Need for security, security approaches-Trusted systems, security models, security management practices, principal of security-Confidentiality, Authentication, Integrity, Non-repudiation, Access control, Availability, Ethical and legal issues, types of attacks, program that attack-Virus, Worm, Trojan Horse, Specific Attack- Sniffing and spoofing, phishing, pharming, Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding. VPN: the basic of Virtual Private Networks.

UNIT-II
Internet security protocols: Introduction, basic concepts, Secure Socket Layer(SSL), Transport Layer Security(TLS), Secure hyper text transfer protocol, Secure Electronics Transaction(SET), SSL Vs SET, 3-D Secure Protocol, E-mail Security, Wireless application Protocol, Security in GSM, Security in 3G ,4G, and 5G, IEEE 802.11 Security.

Indian IT ACT Page 3 of 23 2000. Cyber security initiatives in India - Cyber security strategies and policies

Textbooks:

1. Atul Kahate: Cryptography and Network Security
2. William Stallings, Cryptography and Network Security: Principles and Practice, ISBN 0131873164, 4/e
3. Bruce Schneier, Applied Cryptography (ISBN 0471128457), 2/e

References Books:

1. Wenbo Mao, Modern Cryptography: Theory and Practice, Prentice Hall, 2004
2. Richard A. Mollin, An Introduction to Cryptography, Chapman and Hall/CRC, 2001.
3. A. Menezes, P. Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1997.
4. Thomas H. Barr, Invitation to Cryptography, Prentice Hall, 2002.
5. Richard J. Spillman, Classical and Contemporary Cryptology, Prentice Hall, 2005.

Semester III

Course Code: MCS-601 **Course Title: Data Mining and Knowledge Discovery**

Credit: 4 **LTP: 4-0-0**

Course objectives:

1. To introduce the concepts of Data warehousing and Data mining and its role in the process of Knowledge Discovery in large databases.
2. To explain the concepts of multidimensional data models and Online Analytical Processing.
3. To explain how to design and implement a data warehouse for reasonable sized databases.
4. To explain the working of various algorithms for Mining Association Rules, Clustering, Classification and Sequence Mining
5. To introduce the concepts of Web Mining and Text Mining

Course outcomes:

After successful completion of the course, the students will be able to

1. Learn the concepts of Data Warehouse, Data Mining and its role in the process of Knowledge Discovery in databases.
2. Design and implement reasonable sized data warehouse using multidimensional data models and OLAP operations and tools.
3. Learn thoroughly the principle and working of the Association Rule Mining algorithms and apply it on datasets to discover association rules.
4. Learn the principle and working of leading Clustering algorithms and apply it on datasets to discover cluster and Clustering rules.
5. Learn the principle and working of leading classification algorithms and apply it on datasets.
6. Learn the principle and working of sequence mining algorithms.
7. Learn the concepts of Text and Web Mining.

UNIT-I:

Introduction to Data Mining and data Warehousing, What is Data Warehouse, Definition, Need for data Warehouse, DBMS vs Data Warehouse, Multi-dimensional data Model, Data Cubes, Data Warehouse Schema, stars, snowflakes, and fact constellations, data warehousing architecture and process, Warehouse server, Metadata, Data Warehouse back end process, Data Warehouse physical design – partitioning, indexing, integrity constraints, materialized views, Data Warehouse construction – data extraction, transformation, loading and refreshing.

UNIT-II:

OLAP technique for Data Warehouse, OLAP architecture, operations and OLAP engine, SQL extensions for OLAP, types of OLAP servers, 3-tier Data Warehouse architecture, Data warehouse implementation and data warehousing back-end tools.

Fundamentals of Data Mining, Definitions, KDD vs Data Mining, Data Mining Functionalities, Data Mining techniques, DBMS vs Data Mining, Data Mining techniques, Classification of Data Mining problems, Major issues and challenges of Data Mining, Data Mining tools and Applications.

UNIT-III:

Association Rule Mining in large data bases, Definition and types of Association rules, Association Rule Mining Algorithms: *A priori, Partition, Pincer-Search, FP Tree Growth Algorithms*. Discussion on different Algorithms, *Incremental Algorithm, Border Algorithm*, generalized Algorithm, generalized Association Rule Mining, and Association Rules with item set constraints. Recent advances in Association rule mining.

UNIT-IV:

Clustering techniques: Introduction, clustering paradigms, Categorization of major clustering methods, partitioning algorithms, *k-Means algorithm, k-medoid algorithms: PAM, CLARA, CLARANS, Hierarchical Clustering algorithms: DBSCAN and BIRCH, Categorical Clustering Algorithms: STIRR and ROCK,* Recent advances in Clustering.

UNIT-V:

Classification and prediction: Issues regarding classification and prediction, Classification by Decision tree Induction, Bayesian Classification, Classification by back propagation, Other Classification methods, Prediction, Classifier accuracy.

Overview of Advanced data mining techniques: WEB Mining, Text Mining and Text Clustering, Spatial Mining, Spatial and Temporal data mining.

Text Books:

1. Data Mining Techniques: Arun Kumar Pujari, Universites Press, Third Edition, 2013 or latest edition

2. Data Mining: Concepts and Techniques: Jiawei Han and Micheline Kamber, Morgan Kaufmann Publishers, Third edition, 2011 or latest edition.

Reference Books:

1.Data Mining: Introductory and Advanced Topics: Margaret H Dunham, Pearson Education, 2008 or latest edition.

2. Introduction to Data Mining: Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education, 2009 or latest edition.

Course code: MCS-602 **Course Title: Programming in Java**

Credit: 4 **LTP: 4-0-0**

Course Objectives:

1. To explain how to create a class and how to implement the 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parameterized constructor, single level inheritance, multi-level inheritance using super and without super keyword using java programming.
2. To explain the implementation of exception handling using try, catch, throw, throws, and finally methods, create single thread, multi- thread using thread class and using runnable interface and also implement the join, isalive method along with priority of a thread using java programming.
3. To explain the implementation of event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class using java programming.

Course Outcomes:

1. Learn the implementation of 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parameterized constructor, single level inheritance, multi-level inheritance using super and without super keyword using java programming.
2. Learn the implementation of exception handling using try, catch, throw, throws, and finally methods, create single thread, multi- thread using thread class and using runnable interface and also implement the join, isalive method along with priority of a thread using java programming.
3. Learn the implementation of event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class using java programming.

UNIT-I:

Introduction to Java, Basic Features, Java Virtual Machine Concepts, A Simple Java Program, Primitive Data Type and Variables, Java Keywords, Integer and Floating-Point Data Type, Character and Boolean Types, Declaring and Initialization Variables, Type casting, Java Operators, Expressions, control statements, Arrays.

UNIT-II:

Class Fundamentals, Creating objects ,Assigning object reference variables ,Introducing Methods, Method overloading, Static methods, Constructors, overloading constructors, This Keyword, Using Objects as Parameters, Argument passing, Returning objects ,Method Overriding, Garbage Collection, The Finalize ( ) Method, Inheritance Basics, Access Control ,Multilevel Inheritance, Abstract Classes ,Polymorphism ,Final Keyword, Package, Defining Package, CLASSPATH, Package naming, Accessibility of Packages, Using Package Members, Interfaces, Implementing Interfaces, Interface and Abstract Classes.

UNIT-III:
Exception Handling-try, catch, throw, throws, Multithreaded Programming- Extands Thread class, Runnable interface, join and is alive method, I/O in Java, Text Streams, Buffered Stream, Print Stream, Random Access File, The String Class, String, Buffer Class and Methods.

UNIT-IV:
Applets Programming, Building User Interface with AWT, Swing-based GUI, Layouts and Layout Manager, Container.

UNIT-V:
Event handling – Text filed, Button, Choice List, Radio button, Text area, Java Database Connectivity, Establishing A Connection, Transactions with Database.

Text books/references:

1. Timoth Budd, An Introduction to Object Oriented Programming, Addison Wesley Publishing company (for UNIT--I).
2. Herbert Schildt, The complete Reference, Tata McGraw Hill Publishing company
3. Patrick Naughton and Herbert Schildt, JAVA: the complete Reference, Tata McGraw-Hill Publishing company

Course Code: MCS-603 **Course Title: (A) Wireless & Mobile Computing**

Credit:4 **LTP:4-0-0**

Course Objectives
This course will help students understand the fundamentals of wireless and mobile computing. Also, it gives insight into how technologies evolve over time and recent development in the area.

Course Outcome
After completing this course, student will be able to understand

1. Fundamental of Wireless and Mobile Computing
2. The concept of Modulation and Propagation techniques.
3. The various medium access techniques used in wireless computing
4. Recent advancement in wireless networking.

UNIT I Fundamental of Wireless and Mobile Computing
Fundamentals of Wireless Communication Technology, Mobile computing vs wireless networking, characteristics and Applications, Generations of Wireless Communication- 2G, 2.5G, 3G, 4G and 5G. Introduction to Mobile Computing: Emerging Technologies, GSM, SMS, GPRS, EDGE.

UNIT-II System Design Fundamental
Frequency reuse, channel assignment strategies, handoff strategies-prioritizing handoffs, practical handoff considerations, interference and system capacity- Co-channel interference and system capacity, channel planning for wireless systems, adjacent channel interference, power control for reducing interference.

UNIT-III Radio Propagation mechanisms
Characteristics of the Wireless Channel- Large scale path loss — Path loss models: Free Space and Two-Ray models, Basic propagation mechanisms-reflection, diffraction, scattering, Outdoor propagation model- Longley rice model, okumura model, Indoor Propagation Model- Partition losses (same floor), partition losses between floors. Small scale fading- Factor influencing small scale fading, Types of small scale fading. doppler shift. Parameters of mobile multipath channels.

UNIT-IV: Multiple Access techniques
Introduction to Multiple Access, FDMA, TDMA, CDMA, SDMA. Capacity of cellular systems- capacity of cellular CDMA, Capacity of CDMA with Multiple Cells, Capacity of Space division multiple access.

UNIT-V: Recent Advances Wireless Networks
Introduction, wireless communication protocols in technologies such as 5G, RFID, WiFi-Direct, Li-Fi, LTE, and 6LoWPAN.

Text Book:

1. Rappaport, T. S. (2010). *Wireless communications: Principles and practice, 2/E*. Pearson Education India.

Reference Book:
1.A. Molisch, "Wireless Communications," Wiley, 2005Haykin & Moher, "Modern Wireless Communications" Pearson 2011 (Indian Edition)
2.J. G. Proakis, "Digital Communications," McGraw Hill
3.A. Goldsmith, "Wireless Communications," Cambridge Univ Press, 2005
4.D. Tse and P. Viswanath, "Fundamentals of Wireless Communications," Cambridge Univ Press, 2005
5. Prasant Kumar Pattnaik, Rajib Mall, "Fundamentals of Mobile Computing", PHI Learning 2012
6. Murthy, C. Siva Ram, and B. S. Manoj. Ad hoc wireless networks: Architectures and protocols, portable documents. Pearson education, 2004.

Course Code: MCS-603 **Course Title: (B) Mobile & Ad-Hoc Networks**

Credits: 4 **LTP: 4 – 0 – 0**

Course Objectives:

1. Learn Ad hoc network fundamentals

2. Understand the MAC and different IEEE standards.
3. Understand the different routing protocols.
4. Understand the transport layer and security issues possible in Ad hoc and Sensor networks
5. Learn how to integrate ad-hoc for 4G and 5G at the cross-layer level.

Course Outcomes:

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

1. Know the basics of Ad hoc networks and Wireless Sensor Networks
2. Apply the knowledge to identify appropriate physical and MAC layer protocols
3. Apply this knowledge to identify the suitable routing algorithm based on the network and user requirement
4. Aware of the potential security issue associated with sensor networks and ad-hoc networks at the transport layer.
5. Understand the cross-layer design and integration of ad hoc for 4G & 5G

UNIT-I

INTRODUCTION: Introduction to Ad hoc Networks – Definition, Characteristics, Applications, Wireless channel, Ad hoc Mobility Models: - Indoor and outdoor models.

UNIT-II

MEDIUM ACCESS PROTOCOLS:MAC Protocols, Design Issues, Goals and classification, Contention based protocols- with reservation, Scheduling algorithms, protocols using directional antennas. IEEE standards: 802.11a, 802.11b, 802.11g, 802.15.

UNIT-III

NETWORK PROTOCOLS: Routing Protocols, Design issues, Goals and Classification. Proactive Vs Reactive routing, Unicast Routing Algorithms, Multicast Routing Algorithms, Hybrid Routing Algorithm, Energy Efficient Routing Algorithm, Hierarchical Routing, QoS aware Routing.

UNIT-IV

END-END DELIVERY AND SECURITY: Transport layer, Issues in Designing, Transport layer classification, Ad hoc Transport Protocols, Security issues in Ad hoc Networks: issues and challenges, Network 54 security attacks, Secure routing protocols. Implementation of Ad hoc networks, Performance Analysis

UNIT-V

CROSS LAYER DESIGN AND INTEGRATION OF AD HOC FOR 4G & 5G: Cross layer Design: Need for cross layer design, cross layer optimization, parameter optimization techniques, Cross layer cautionary perspective. Integration of ad hoc with Mobile IP networks, WAP.

TEXT BOOKS:

1. C.K.Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Pearson, 2001
2. C.Siva Ram Murthy and B.S.Manoj, Ad hoc Wireless Networks Architectures and protocols, 2nd edition, Pearson Education. 2007
3. Charles E. Perkins, Ad hoc Networking, Addison – Wesley, 2000

REFERENCES:

1. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan Stojmenovic, Mobile Adhoc networking, Wiley-IEEE press, 2004.
2. Mohammad Ilyas, The Handbook of Adhoc Wireless Networks, CRC press, 2002.

Course Code: MCS-603 **Course Title: (C) Natural Language Processing**

Course Credit-4 **L-T-P: 4-0-0**

Course Objectives

1. To introduce and familiarize the students with the fundamental concepts and techniques of NLP.
2. To introduce and familiarize the students with state-of-the-art research in the field of NLP such as Machine Translation, Sentiment and Opinion Mining, Chat-bots, Dialogue Agents, Chat GPT.

Course Outcome

1. Understand Stages of NLP, Tokens, Stemming and Lemmatization in NLP, Morphology and morphological diversity of Indian Languages.
2. Understand Language modelling, Parsing, PCFG, Ambiguity and ambiguity resolution.
3. Understand POS Tagging, Word embedding, Wordnet and WSD
4. Will have gained sufficient insight to undertake Projects in Machine Translation, WSD, Sentiment analysis, implement chat-bot and Dialogue Agents.

Unit-I

Introduction to NLP, Stages of NLP; Basic Text Processing- tokens, sentences, paragraphs; Statistical properties of words; Morphology fundamentals, Morphology Diversity of Indian Languages, Finite state machine based morphology; Word boundary detection; Argmax based computation; Stemming and Lemmatization; Approaches to NLP

Unit-II

Language Modelling- N gram language model, Evaluation of language models, Basic smoothing ; Sequence Labelling and noisy channel, argmax based computation in a noisy channel; Syntax- Theories of Parsing, Parsing Algorithms, PCFG ; Ambiguity and ambiguity resolution; TF-IDF Classification, Document similarity measure- Cosine and cluster measure

Unit-III

Introduction to POS Tagging, HMM for POS Tagging, Viterbi decoding for HMM, Baum Welch Algorithm, Maximum Entropy Models ; Precision, Recall, F-score; Information retrieval and extraction Word vectors, GloVe/Word2Vec model, word embeddings; EM algorithm in NLP; Wordnet.

Unit-IV

WSD and WSD algorithms ; Machine Translation: Encoder and Decoder model of Neural Machine Translation, RNN based Translation, Attention based Translation, LSTM, Typical NMT architecture and model for multi language Translation

Unit-V

Sentiment Analysis and Opinion Mining; Conversation Modelling, Chat-bots, dialog agents, Question Processing; Chat GPT

Textbooks:

1. Speech and Language Processing, Jurafsky & Martin, Pearson
2. Foundations of Statistical Natural language processing, Manning & Schutze, MIT Press

3. Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, Steven Bird, Ewan Klein, Edward Loper , O'Reilly https://www.nltk.org/book/

Reference:

1.Machine Translation, Pushpak Bhattacharya, CRC Press

2.Dipanjan Sarkar, Text Analytics with Python, Dipanjan Sarkar (Apress/Springer, 2016)

3.Linear Algebra and learning from Data, Gilbert Strang, Wellesley Cambridge Press

4. Introduction to Probability,Bertekas, John Tsitsklis, Athena Scientific(online ver. available)

Course Code: MCS-603 **Course Title: (D) Software Engineering**

Credit: 4 **LTP: 4-0-0**

Course Objectives:

1. To introduce the various stages of SDLC.
2. To introduce the various process models.
3. To introduce the software quality, reliability and configuration management

Course Outcomes:

Upon successful completion of this course, the student shall be able to:

1. Understand the various stages of SDLC, the various process models, software quality, reliability and configuration management.
2. Develop alternative solutions for the system.
3. Implement, test and validate a system designs and develop the software

UNIT-I:

Software: Importance of software, Characteristics, Components, Applications of Software, Software Myths. Planning and Management of software Project**:** People, problem and process, measures, matrices and indicators, matrices for software quality, scooping, software project estimation, make-buy decision, software acquisition. Life Cycle Models: linear sequential (Waterfall) model, Iterative models, RAD Model, Prototyping Model, Spiral Model.

UNIT-II:

Project scheduling and tracking: tasks/work breakdown structures, Activity Network and CPM, Gantt and PERT Charts, timeline chart, CASE tools. Requirements Elicitation: Interviews, Brainstorming, FAST Requirement Analysis**:** Data flow diagrams, behavioral models, mechanics of structured analysis, ER diagrams, data dictionary, Software Prototyping. Requirement Documentation**:** Nature, characteristics and organization of SRS and SRS reviews. Software Project Planning**:** LOC, Function count, Empherical and Heuristic Estimation Techniques: COCOMO, Intermediate COCOMO, Complete COCOMO and COCOMO-II models. Software risk management**:** Risk Identification, Risk assessment, Risk monitoring Risk Containment.

UNIT-III:

Software Design: Conceptual and Technical Designs, Objectives of Design Modularity**:** Module Coupling, Module Cohesion, Relation between cohesion and coupling. Design Strategy**:** Bottom- up, Top-Down, Hybrid Design. Function oriented design**:** design notations, Functional Procedural layers, DFD, Flowchart, Structure charts, Transform and transaction analysis. Object Oriented Design**:** Basic mechanism, concepts, advantages of OOD, unified modeling language (UML).

UNIT-IV:

Software testing and testing strategies : Software testing fundamentals, test case design, white-box, black-box testing, control structure testing, strategic approach to testing, strategic issues, Unit testing, integrated testing, validation testing, system testing.

UNIT-V:

Software quality, reliability and configuration management: Software quality concepts, Software quality assurance (SQA) and approaches, SQA plan. Software Reliability: ISO 9000 and SEI standards for software. Software configuration management (SCM): base lines, scan process, version control, change control, SCM audits.

Text Books:

1. Roger S. Pressman: Software Engineering, A Practitioner's Approach, 7th Ed., Tata McGraw Hill pub.
2. Rajib Mall: Fundamentals of Software Engineering, 5th Ed., PHI pub

References Books:

1. Pankaj Jalote: An Integrated Approach of Software Engineering (Galgotia), 3rd Ed. Springer pub
2. K. K. Aggarwal, Yogesh Singh: Software engineering, 3rd Ed., New Age International publishers
3. F. Tsui, O. Karam, B. Bernal: Essentials of Software Engineering, 4th Ed., Jones & Bartlett pub

Course Code: MCS-603 **Course Title: (E)Distributed Computing**

Credit: 4 **LTP: 4-0-0**

Course Objectives

1. To introduce the concept of Distributed Computing, its models and principles.
2. To explain the concepts and working of IPC, RPC and Distributed Shared memory Systems.
3. To explain the concepts of Distributed Mutual Exclusion, event handling and Deadlock.
4. To explain the concepts of resource management -load balancing, load sharing and Task assignment techniques
5. To explain the concepts of process management in distributed systems and distributed file
systems.
6.To explain concept and working of distributed file systems.
7. To illustrate the concept of distributed naming systems and overview of security in distributed systems.

Course Outcomes

After successful completion of the course, the students will be able to
1. Learn Concept and models of distributed systems.
2. Learn how IPC, RPC and Distributed shared memory systems work.
3. Understand the concepts of event handling, Mutual Exclusion and Deadlock and related algorithms.
4. Understand distributed resource and process management techniques and related mechanisms
- load sharing, load balancing, task assignment, process migration etc.
5. Understand about the distributed file systems and naming systems.
6. Familiarise with security techniques in distributed systems.

UNIT-I:

Fundamentals: Introduction, Models and Features, Concept of distributed operating system, Issues in design of a distributed operating system, Client Server Computing.

Message Passing: Good message passing system, IPC, Synchronization, Buffering, Multi datagram messages, Encoding & decoding techniques, Process addressing, Failure handling, Group communication; Remote procedure calls (RPC) - Models, Communication protocols, RPC, Lightweight RPC.

UNIT-II:

Distributed Shared Memory: Architecture, Thrashing, Granularity, Advantages.Synchronization: Introduction, Clock Synchronization, Event handling, Mutual Exclusion; Deadlock – Conditions, Avoidance, Prevention, Recovery.

UNIT-III:

Resource & process Management: Features of a good scheduling algorithm, Task assignment approach, Load balancing & load sharing approach, Introduction to process management, Process migration, Threads.

UNIT-IV:

Distributed Files Systems: Introduction, Features, Models, Accessing models; sharing Semantics & caching schemes, replication, Fault Tolerance, Atomic transactions.Distributed File Servers, Distributed Real Time System

UNIT-V:

Distributed Database, Concurrency Control in Distributed Database, Naming: Introduction, Features, Fundamental Terminologies & concepts, System oriented names, Human oriented names, Name caches.

Security: Potential attacks to computer system, Cryptography, Authentication, digital signatures, Access Control.

Textbooks:

1. Sinha Pradeep K., Distributed operating Systems, Concepts & design, PHI.
2. Tanenbaum Andrews S., Distributed Operating System, Pearson Education.

Reference Books:

1. Coulouris George, Dollimore Jean, Kindberg Tim, Distributed Systems, Concepts & design, Pearson.
2. Silberschatz& Galvin, Operating System Concepts, John Wiley, 5th Edition.

Course Code: MCS-603 **Course Title: (E) Pattern Recognition**

Credit: 4 **LTP: 4-0-0**

Course Objectives:

1. Introduces the different components of a pattern recognition system and different techniques in statistical pattern recognition
2. Introduces basic concept of classifier and major techniques of classifier design.
3. Introduces various types of possible error in classification and different error assessment techniques.
4. Explain important feature extraction techniques and introduce the basic concept of clustering technique
5. To introduce advance algorithms and different applications in the field of pattern recognition

Course Outcomes:

Upon completion of the course student will able to

1. Understand the basic concepts and algorithms of pattern recognition.
2. Apply parametric and no-parametric classifiers
3. Perform error assessment to improve the performance of classifier.
4. Extract features from raw data and apply unsupervised data clustering on the extracted data.
5. Identify emerging trends and directions in the field of pattern recognition

UNIT-I:

Introduction: Examples; The nature of statistical pattern recognition; Three learning paradigms; The sub-problems of pattern recognition; The basic structure of a pattern recognition system; Comparing classifiers. Bayes Decision Theory: General framework; Optimal decisions; Classification; Simple performance bounds.

UNIT-II:

Learning - Parametric Approaches: Basic statistical issues; Sources of classification error; Bias and variance; Three approaches to classification: density estimation, regression and discriminant analysis; Empirical error criteria; Optimization methods; Failure of MLE; Parametric Discriminant Functions: Linear and quadratic discriminants; Shrinkage; Logistic classification; Generalized linear classifiers; Perceptrons; Maximum Margin; Error Correcting Codes.

UNIT-III:

Error Assessment: Sample error and true error; Error rate estimation; Confidence intervals; Resampling methods; Regularization; Model selection; Minimum description length; Comparing classifiers Nonparametric Classification: Histograms rules; Nearest neighbor methods; Kernel approaches; Local polynomial fitting; Flexible metrics; Automatic kernels methods

UNIT-IV:

Feature Extraction: Optimal features; Optimal linear transformations; Linear and nonlinear principal components; Feature subset selection; Feature Extraction and classification stages, Unsupervised learning and clustering, Syntactic pattern recognition, Fuzzy set Theoretic approach to PR.

UNIT-V:

Margins and Kernel Based Algorithms: Advanced algorithms based on the notions of margins and kernels Applications of PR: Speech and speaker recognition, Character recognition, Scene analysis.

Textbooks:

1. Theodoridis & Koutroumbas, Pattern Recognition, Academic Press
2. R. Duda, P. Hart and D. Stork. Pattern Classification. 2nd Ed, Wiley, 2001.

Course Code: MCS-603 **Course Title: (F) Operation Research and Optimization**

Credit:4 **LTP:4-0-0**

Course Objectives:

1. To understand the methodology of problem solving and formulate linear programming problem.

2. To Understand and analyzing different situations in the industrial/business scenario involving limited resources and finding the optimal solution within constraints.

Course Outcomes:

1. Convert the decision making problem into a mathematical model and solve by using appropriate OR techniques.

2. Understand variety of problems such as assignment, transportation, travelling, salesman etc.

3. Model competitive real-world phenomena using concepts from game theisy. Analyse pure and mixed strategy games.

UNIT-I:

Introduction, convexity and related results, linear programming problem, Solution by graphical and Simplex method, theory of Simplex method, optimality condition, Duality and Fundamental theorem of duality.

UNIT-II:

Two phase Simplex method, Big M method, Revised Simplex Method, Decomposition principle, Sensitivity Analysis, Parametric linear programming.

UNIT-III:

Study of transportation problem, Method for finding initial solutions (North-west corner method, Least cost method, Vogel's Approximation method), Modi method for optimum solution, Assignment problems, Hurgerian method and traveling salesman problem.

UNIT-IV:

Introduction to game theory, Maximum-minimum principle, games without saddle point, reduction to LPP, Networks Scheduling by PERT and CPM, Project cost, Time cost Optimization algorithm, Probability in PERT analysis.

UNIT-V:

Nonlinear Programming: Convex and non convex programming, Kun Tucker conditions for constrained optimization, Quadratic programming.

Integer programming: Branch and bound technique, Gomorys cutting plane method.

Text Books/References:

1. H.A. Taha, Operations Research, An introduction, PHI, 2004.
2. J.K. Sharma, Operation Research, Theory and Applications, 4th edition, Mcmillan, 2009.
3. S.I. Gass, Linear Programming, Methods and Applications, 5th edition, Dover publications, 2013.
4. K.P.P. Chong, S.H. Zak: An introduction to Optimization, John Welly &Sons , 2001.

Course Code: MCS-603 **Course Title: (G) Digital Image Processing**

Credit:4 **LTP:4-0-0**

Course objectives:

1. To introduce the concepts of Digital Image Processing and fundamental techniques.

2. To explain the concepts of Digital Image Transforms

3.To explain image enhancement and Color image processing techniques

4. To explain image restoration degradation model

5. To explain Image segmentation techniques

Course outcomes: After successful completion of the course, the students will be able to
1. Learn image conversion and image transformation techniques.

2. Apply image enhancement techniques

3. Apply color image processing techniques

4. Learn image restoration degradation models.

5. Apply various image processing techniques for real time applications

UNIT-I:

Digital image fundamentals - Concept of gray levels. Gray level to binary image conversion. Sampling and quantization.Relation ship between pixels.Imaging Geometry. Image Transforms 2-D FFT , Properties. Walsh transform, Hadamard Transform, Discrete cosine Transform, Haar transform, Slant transform, Hotelling transform.

UNIT-II:

Image enhancement Point processing.Histogramprocessing.Spatialfiltering.Enhancement in frequency domain, Image smoothing, Image sharpening.

UNIT-III:

Colour image processing :Psedocolour image processing, full colour image processing. Image compression Redundancies and their removal methods, Fidelity criteria, Image compression models, Source encoder and decoder, Error free compression, Lossy compression.

UNIT-IV:

Image Restoration Degradation model, Algebraic approach to restoration, Inverse filtering, Least mean square filters, Constrained Least Squares Restoration, Interactive Restoration.

UNIT-V:

Image segmentation Detection of discontinuities. Edge linking and boundary detection, Thresholding, Region oriented segmentation.

TEXT BOOKS:

1. Digital Image processing – R.C. Gonzalez & R.E. Woods, Addison Wesley/ Pearson education, 2nd Education, 2002.

REFERENCES:

1. Fundamentals of Digital Image processing – A.K.Jain , PHI.

2. Digital Image processing using MAT LAB – Rafael C. Gonzalez, Richard E Woods and Steven L. Edition, PEA, 2004.

3. Digital Image Processing – William K. Pratt, John Wilely, 3rd Edition, 2004.

4. Fundamentals of Electronic Image Processing – Weeks Jr., SPIC/IEEE Series, PHI.

Course Code: MCS-604 Course Title: (a) Lab on Data Mining and Knowledge Discovery

Credit: 4 **LTP: 4-0-0**

Course Objectives:

1. To design and implement reasonable sized data warehouse by using multidimensional data models and OLAP Operations in SQL/C/C++/Java/Python
2. To implement the Apriori, Partition, Pincer-Search and FP Tree Growth algorithms in transactions databases for discovery of Association Rules in C/C++/Java/Python
3. To implement Clustering algorithms: k-Means, PAM, DBSCAN and two categorical clustering algorithms using C/C++/Java/Python
4. To implement ID 3 and algorithm for classification
5. To introduce the students to open source data mining systems like Weka, illimine etc. and open source big data repositories for Data Mining and Machine Learning through self learning

Course outcomes:

After successful completion of the course students will be able to

1. Write programs and implement reasonable sized data warehouse using multidimensional data models and OLAP operations using SQL/C/C++/JAVA/PYTHON.
2. Write programs for Apriori, Partition and FP Tree Growth algorithms and implement in C/C++/JAVA/PYTHON.
3. Write programs and implement for the k-Means, PAM, DBSCAN and two categorical Clustering algorithms in C/C++/JAVA/PYTHON
4. To write programs in C/C++/JAVA/PYTHON and implement the ID 3 and CART algorithms for classification.
5. Students will install and work with open source data mining systems like WEKA, illimine etc. by using open source big data repositories for Data Mining and Machine Learning

Detail Syllabus for Laboratory Programming and Experiments for Lab on Data Mining and Knowledge Discovery

Problems and various algorithms related to Data Ware Housing and Data Mining should be solved and implemented by using the Programming languages C/C++/JAVA/PYTHON /SQL/ORACLE as well as various tools for Data Ware Housing and Data Mining. Following are some tasks and algorithms of Data Ware Housing and Data Mining for laboratory programming assignments but the assignments should not be limited to these only.

(i) Data ware house design and implementation, Data Cube Design and Implementation, Implementation of the OLAP Operations: Write programs and implement reasonable sized data warehouses using multidimensional data models and OLAP operations using SQL/C/C++/JAVA/PYTHON.

(ii) Implementation of Apriori, Partition, Pincer-Search and FP Growth Algorithms in transactions databases for discovery of Association Rules in C/C++/ JAVA/PYTHON

(iii) Implementation of k-Means, k-medoid algorithms and any one of each of the hierarchical and categorical clustering algorithms: Write programs and implement for the k-Means, PAM, DBSCAN and one categorical Clustering algorithm in C/C++/JAVA/PYTHON

(iv) To write programs in C/C++/JAVA/PYTHON and implement the ID 3 and CART algorithms for classification.

(v) Implementation of one classification algorithm and the Page Rank algorithm.

(vi) Students will install and work with open source data mining systems like WEKA, illimine etc. by using open source big data repositories for Data Mining and Machine Learning.

Course Code: MCS-604 **Course Title: (b)MCS- Lab on Programming in Java**

Course Objectives:

1. To create a class and implement 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parametrized constructor, single level inheritance, multi-level inheritance using super and without super key word

2. To implement exception handling using try, catch, throw, throws, and finally methods, create single thread , multi- thread using thread class and using runnable interface and also implement the join, isalive method along with priority of a thread.
3. To implement event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class.

Course outcomes:

After successful completion of the course students will be able to

1. Write a program to implement 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parametrized constructor, single level inheritance, multi-level inheritance using super and without super key word
2. Write a program to implement exception handling using try, catch, throw, throws, and finally methods, create single thread, multi- thread using thread class and using runnable interface and also implement the join, isalive method along with priority of a thread.
3. Write a program to implement event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class.
4. Develop a GUI application of your choice that includes label, button, check box, radio button, list, choice, text field, text area along with JDBC connectivity with the database and also implement the event handling on desire components.

Programming problems should be solved by using the high level and Object Oriented Programming language JAVA (preferably on windows/ Linux environment on a network). Following are some areas of JAVA for laboratory programming assignments but the assignments should not be limited to these only.

LIST OF EXPERIMENTS

1. Write a java program to implement 1-D,2-D array. Implement the matrix addition, subtraction, multiplication, and division operations.
2. Design a class for Complex numbers in Java. In addition to methods for basic operations on complex numbers, provide a method to return the number of active objects created.
3. Design a Date class similar to the one provided in the java.util package.
4. Develop with suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc.
5. Design a simple test application to demonstrate polymorphism, method overloading, default constructor, parametrized constructor, single level inheritance, multi-level inheritance using super and without super key word.
5. Develop two different classes that implement this interface, one using array and the other using linked-list. Provide necessary exception handling in both the implementations.
6. Write a Java program to implement exception handling using try, catch, throw, throws, and finally methods.
7. write a java program to create single thread , multi- thread using thread class and using runnable interface and also implement the join, isalive method along with priority of a thread.
8. Develop a simple paint-like program that can draw basic graphical primitives in different dimensions and colors. Use appropriate menu and buttons.
9. Develop a scientific calculator using even-driven programming paradigm of Java.

10. Implement event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class.

11. Develop a GUI application of your choice that includes label, button, check box, radio button, list, choice, text field, text area along with JDBC connectivity with the database and also implement the event handling on desire components.

Semester IV

Course Code: MCS-651 **Course Title: Data Science**

Credits: 4 **LTP: 4 – 0 – 0**

Course Objectives:

1. To explain the basics of data science and syntax, execution, and programs in R.
2. To explain the use of matrices and arrays in R.
3. To explain the use of non-numeric values in vectors.
4. To explain the use of data frame and list in R.
5. To explain the graph plotting using ggplot2.

Course Outcomes:

Upon completion of this course, the students will be able to:

1. Understand the concept of data science and execute basic syntax and programs in R.
2. Perform the Matrix operations using R built in functions.
3. Apply non numeric values in vectors.
4. Create the list and data frames.
5. Exploit the graph using ggplot2.

UNIT – I OVERVIEW OF DATA SCIENCE & R

Introduction to Data Science – Evolution of Data Science – Data Science Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues.

History and Overview of R- Basic Features of R-Design of the R System- Installation of R-Console and Editor Panes- Comments- Installing and Loading R Packages- Help Files and Function Documentation Saving Work and Exiting R- Conventions- R for Basic Math-Arithmetic- Logarithms and Exponentials E-Notation- Assigning Objects- Vectors-Creating a Vector- Sequences, Repetition, Sorting, and Lengths- Sub setting and Element Extraction- Vector-Oriented Behaviour.

UNIT – II MATRICES AND ARRAYS

Defining a Matrix – Defining a Matrix- Filling Direction- Row and Column Bindings-Matrix Dimensions Sub setting- Row, Column, and Diagonal Extractions- Omitting and Overwriting- Matrix Operations and Algebra- Matrix Transpose- Identity Matrix- Matrix Addition and Subtraction- Matrix Multiplication Matrix Inversion-Multidimensional Arrays- Subsets, Extractions, and Replacements.

UNIT – III NON-NUMERIC VALUES

Logical Values- Relational Operators- Characters- Creating a String- Concatenation-Escape Sequences Substrings and Matching- Factors- Identifying Categories- Defining and Ordering Levels- Combining and Cutting.

UNIT – IV LISTS AND DATA FRAMES

Lists of Objects-Component Access-Naming-Nesting-Data Frames-Adding Data Columns and Combining Data Frames-Logical Record Subsets-Some Special Values-Infinity-NaN-NA-NULL Attributes-Object-Class-Is-Dot Object-Checking Functions-As-Dot Coercion Functions.

UNIT – IV BASIC PLOTTING

Using plot with Coordinate Vectors-Graphical Parameters-Automatic Plot Types-Title and Axis Labels Color-Line and Point Appearances-Plotting Region Limits-Adding Points, Lines, and Text to an Existing Plot-ggplot2 Package-Quick Plot with qplot-Setting Appearance Constants with Geoms-- READING AND WRITING FILES- R-Ready Data Sets- Contributed Data Sets- Reading in External Data Files- Writing Out Data Files and Plots- Ad Hoc Object Read/Write Operations.

TEXT BOOKS

1. Tilman M.Davies,"THE BOOK OF R - A FIRST PROGRAMMING AND STATISTICS" Library of Congress Cataloging-in-Publication Data,2016.

REFERENCE BOOKS

1. Roger D. Peng,"R Programming for Data Science"Lean Publishing, 2016.
2. Hadley Wickham, Garrett Grolemund," R for Data Science",OREILLY Publication,2017
3. Steven Keller, "R Programming for Beginners", CreateSpace Independent Publishing Platform 2016.
4. Kun Ren ,"Learning R Programming", Packt Publishing,2016

E – BOOKS

1. https://web.itu.edu.tr/~tokerem/The_Book_of_R.pdf

MOOCs

1. https://online-learning.harvard.edu/subject/r
2. https://www.udemy.com/course/r-basics/
3. https://www.datacamp.com/courses/free-introduction-to-r

Course Code: MCS-652 **Course Title: Machine Learning**

Credits: 4 **LTP: 4 – 0 – 0**

Course Objectives:

1. To explain the basics of machine learning and its applications.
2. To explain the use of knowledge in learning.
3. To explain the statistical methods in machine learning.
4. To explain the reinforcement learning and clustering of data.
5. To explain the advanced machine learning techniques.

Course Outcomes:

Upon completion of this course, the students will be able to:

1. Design machine learning models and implement it.
2. Perform the computation parts of any research oriented problems with respect to machine learning.
3. Conceive the idea to start Python programs to solve AI and ML problems.
4. Create and execute ML programs.
5. Extend the application of ML for other advanced areas.

UNIT – I OVERVIEW OF MACHINE LEARNING

Introduction to machine learning, forms of learning, learning from observations, inductive learning, concept learning and the general-to-specific ordering, version spaces and the candidate-elimination algorithm, learning decision trees and ensemble learning, computational learning theory, bootstrapping & cross validation, class evaluation measures, ROC curve, MDL, ensemble methods - bagging, committee machines and stacking, boosting

UNIT – II KNOWLEDGE IN LEARNING & REGRESSION

Logical formulation of learning, knowledge in learning, explanation-based learning (EBL), relevance-based learning (RBL), knowledge-based inductive learning (KBIL), inductive logic programming (ILP)
Linear Regression, Multivariate Regression, Subset Selection, Shrinkage Methods, Principal Component Regression, Partial Least squares, Linear Classification, Logistic Regression, Linear Discriminant Analysis

UNIT – III STATISTICAL LEARNING METHODS

Statistical learning, probability theory, learning with complete data, discrete models: Naïve Bayes models, continuous models: parameter learning, learning Bayes Net structure, learning with hidden variables: expectation-maximization algorithm, Unsupervised clustering, learning mixtures of Gaussian instance based learning, neural networks, kernel machines, support vector machines (SVM)

UNIT – IV REINFORCEMENT LEARNING & CLUSTERING

Introduction to Reinforcement learning, passive reinforcement learning, active reinforcement learning, generalisation in reinforcement learning
Partitioning Clustering, Density-Based Clustering, Distribution Model-Based Clustering, Hierarchical Clustering, Fuzzy Clustering

UNIT – V ADVANCED MACHINE LEARNING

Emergent models of learning, genetic algorithm, classifiers system, genetic programming, artificial life and society based learning, introduction to deep learning, deep learning architectures, Hidden Markhov models, man-machine interfaces

TEXT BOOKS

1. Tom M. Mitchell, Machine Learning, McGraw Hill Education, Indian Edition, 2017
2. S Sridhar and M Vijayalakshmi, Machine Learning, Oxford University Press, 2021

REFERENCE BOOKS

1. Anuradha Srinivasaraghavan and Vincy Joseph, Machine Learning, Wiley Press, 2019
2. Manaranjan Pradhan and U Dinesh Kumar, Machine Learning using Python, IIMB Edition, Wiley Press, 2019
3. Sujit Bhattacharyya and Subhrajit Bhattacharyya, Practical Handbook of Machine Learning, GK Publications, 2021

E – BOOKS

1. https://alex.smola.org/drafts/thebook.pdf
2. https://ai.stanford.edu/~nilsson/MLBOOK.pdf

MOOCs

1. https://onlinecourses.nptel.ac.in/noc23_cs18
2. https://towardsdatascience.com/

Course Code: MCS-653 **Course Title: (A) ARTIFICIAL NEURAL NETWORKS**

Credit: 4 **LTP:4-0-0**

Course Objectives:

1. To understand the biological neural network and to model equivalent neuron models.
2. To understand the architecture, learning algorithm and issues of various feed forward and feedback neural networks.

Course Outcomes: By completing this course the student will be able to:

1. Create different neural networks of various architectures both feed forward and feed backward.
   Perform the training of neural networks using various learning rules.
2. Perform the testing of neural networks and do the perform analysis of these networks for various pattern recognition applications.

UNIT - I

Introduction: A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural Networks

Learning Process: Error Correction Learning, Memory Based Learning, Hebbian Learning, Competitive, Boltzmann Learning, Credit Assignment Problem, Memory, Adaption, Statistical Nature of the Learning Process

UNIT – II

Single Layer Perceptrons: Adaptive Filtering Problem, Unconstrained Organization Techniques, Linear Least Square Filters, Least Mean Square Algorithm, Learning Curves, Learning Rate Annealing Techniques, Perceptron –Convergence Theorem, Relation Between Perceptron and Bayes Classifier for a Gaussian Environment

Multilayer Perceptron: Back Propagation Algorithm XOR Problem, Heuristics, Output Representation and Decision Rule, Computer Experiment, Feature Detection

UNIT - III

Back Propagation: Back Propagation and Differentiation, Hessian Matrix, Generalization, Cross Validation, Network Pruning Techniques, Virtues and Limitations of Back Propagation Learning, Accelerated Convergence, Supervised Learning

UNIT – IV

Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Patter Classification

UNIT – V

Neuro Computation: Domains of Application of Neural Networks – Expert System & Decision Making system, Pattern Recognition, Neuro Controllers and Fuzzy Neuro Controllers, Recurrent Neural Network, Convolutional Neural Network.

TEXT BOOKS:

1. Neural Networks a Comprehensive Foundations, Simon Haykin, PHI edition.

REFERENCE BOOKS:

1. Artificial Neural Networks - B. Vegnanarayana Prentice Hall of India P Ltd 2005
2. Neural Networks in Computer Inteligance, Li Min Fu MC GRAW HILL EDUCATION 2003
3. Neural Networks -James A Freeman David M S Kapura Pearson Education 2004.
4. Introduction to Artificial Neural Systems Jacek M. Zurada, JAICO Publishing House Ed.

Course Code: MCS-653:
Course Title: (B)Advanced Computer Architecture and Parallel Computing

Credit: 4 **LTP: 4-0-0**

Course Objectives

1. To introduce the concept of Parallel Computing architecture, its models and principles.
2. To explain the concepts and working of Interconnection network for parallel Computing Systems.
3. To explain the concepts of Parallel Computing techniques.
4. To explain the concepts and working of Parallel Sorting algorithms
5. To explain the concepts and working of Parallel Search Algorithms

Course Outcomes

After successful completion of the course, the students will be able to

1. Learn the concept of Parallel Computing architecture, its models and principles
2. Learn concepts and working of Interconnection network for parallel Computing Systems
3. Understand the concepts of Parallel Computing techniques.
4.Understand concepts and working of Parallel Sorting algorithms
5. Understand about the concepts and working of Parallel Search Algorithms

UNIT-I:

Advanced Computer Architecture: Introduction to Parallel Processing, Parallel Computer Structures, Pipeline and Array Computers, Multiprocessor Systems, Architectural Classification Scheme.

Interconnection network: Tree, Diamond Network, Mesh, Linear array, Ring, Star, Hypercube, Choral ring, Cube- connected cycles, perfect shuffle network, Torus, PM 21, Butterfly, Mesh of tree, Pyramid, Generalized Hyperbus, Twisted cube, Folded Hypercube, Incomplete Hypercube, Enhanced Incomplete Hypercube, Cross Connection Cube, Banyan Hypercube.

UNIT-II:

Principles of pipeline and Vector-Processing, Multification and Array Pipelines, Design of Pipelined Processors, Data buffering and busing System, Vector Processing Requirements, Pipeline Computers and Vectorization Methods, Architecture of Typical Vector Processors, Vectorization and Optimization Methods.

Structures and Algorithms for Array Processors, SIMD Array Processors, SIMD Interconnection Networks, Typical Parallel Processors, Multiprocessor Architecture, Loosely and tightly coupled Multiprocessor.

UNIT-III:

SIMD Array Processors and SIMD Inter connection networks. Principles of Parallel Computing: Message Passing Parallel Programming, PVM and MPI.Introduction to Pipelined Computations. Parallel Computation Models: PRAM, CRCW, CREW, EREW, Simulating CRCW on CREW & EREW.

UNIT-IV:

Parallel Sorting : Odd – Even transportation sort on Linear Array, Merge Splitting sorting, Quick Sort, Theorem of Odd-Even Merging, Zero- One Principle, Bitonic Sort. Matrix Multiplication: Sequential Matrix Multiplication: Row wise Block – Striped Parallel Algorithm, Cannon's Algorithm,

UNIT-V:

Parallel Search Algorithms: Parallel Depth First Search, Parallel Breadth First Search, Parallel Branch and Bound Search, Parallel Best-First Search. Not smaller-than search, Distributed Real Time System, Data Flow Computer(Static and Dynamic) Architecture, Reduced Instruction Set Computer and Architecture Characteristics.

Text Books/References::

1. K.Hwang and F.A.Briggs : Computer Architecture and Parallel Processing (McGraw Hill)
2. K.Hwang : Super Computer Design and Application (Computer Society Press)
3. Kai Hwang, "Advanced Computer Architecture – Parallelism, Scalability, Programmability", McGraw Hill Inc.
4. V.Rajaraman: Elements of Parallel Computing (PHI). (Latest Edition)
5. Barry Wilkinson and Michael Allen: Parallel Programming: Techniques and Applications (Pearson Education) Latest Edition.
6. AnanthGrama, Anshul Gupta, George Karypis and Vipin Kumar: Introduction to Parallel Computing (Latest Edition), (Pearson Education)
7. M.J.Quinn: Parallel Programming in C with MPI and Open MP: Tata MC Graw Hill.

Course Code: MCS-653 **Course Title: Quantum computing**

Credit :4 **LTP: 4-0-0**

Course Objective

1. To explain the basics of quantum computing and its applications.
2. To explain the use of qubits in computation.
3. To explain the design of quantum circuits.
4. To explain the quantum algorithms.
5. To explain the performance, security, scalability of QC.

Course Outcome
Upon completion of this course, the students will be able to:

1. Analyze the behaviour of basic quantum algorithms
2. Implement simple quantum algorithms and information channels in the quantum circuit model
3. Simulate a simple quantum error-correcting code
4. Prove basic facts about quantum information channels
5. Design new quantum algorithms and implement them.

UNIT – I OVERVIEW OF QUANTUM COMPUTERS

Introduction to Quantum Computers: Properties, Architectures, Difference with Classical Computers. Quantum Superposition and Entanglement, Quantum Bits, Quantum Gates and Circuits, Elementary quantum mechanics, linear algebra for quantum mechanics, Quantum states in Hilbert space, The Bloch sphere, Density operators, generalized measurements.

UNIT – II QUANTUM COMPUTING PRINCIPLES

Quantum Computing Basics and Principles: No cloning theorem & Quantum Teleportation, Quantum correlations, Bell's inequality and entanglement, Schmidt decomposition, Super dense coding.

Introduction to Qiskit Tools, IBM Quantum Composer, IBM Quantum Lab using Qiskit, Programming QC in Python open source platform, Circuit design, Analysis of Quantum Outputs, Quantum Computer Job Execution Process.

Detailed Analysis of Quantum Gates, Circuits, Functions, Basic Programming in QC, New algorithms based on Existing Gates.

UNIT – III QUANTUM ALGORITHMS

Algorithms: Deutsch and Deutsch–Jozsa algorithms, Grover's Search Algorithm, Quantum Fourier Transform, Shor's Factorization Algorithm.

Bernstein–Vazirani algorithm, Simon's algorithm, QAOA, VQE

UNIT – IV PERFORMANCE, SECURITY AND SCALABILITY

Performance, Security and Scalability: Quantum Error Correction, Fault tolerance, Quantum Cryptography, Implementing Quantum Computing: issues of fidelity, Scalability in quantum computing.

UNIT – V QUANTUM COMPUTING MODELS

Quantum Computing Models: NMR Quantum Computing, Semiconductor Spintronics and QED MODEL, Linear & Non Linear Optical QC Models, Lorentz Model, Adiabatic Quantum Computing (AQC) Model, Topological Quantum Computing (TQC) Model, Quantum Turing Machine (QTM), Limits of Models, Future of Quantum Computing Models.

TEXT BOOKS

1. Christine Corbett Moran, Mastering Quantum Computing with IBM QX: Explore the world of quantum computing using the Quantum Composer and Qiskit, Kindle Edition, Packt Publishing, 2019.
2. Vishal Sahni, Quantum Computing, Tata McGraw Hill Education, 2011.

REFERENCE BOOKS

1. Michael A. Nielsen and Issac L. Chuang, Quantum Computation and Information, Cambridge Press, 2002.
2. Riley Tipton Perry, Quantum Computing from the Ground Up, World Scientific Publishing Ltd, 2012.
3. Scott Aaronson, Quantum Computing since Democritus, Cambridge Press, 2013.

E – BOOKS

1. http://mmrc.amss.cas.cn/tlb/201702/W020170224608150244118.pdf
2. https://csis.pace.edu/~ctappert/cs837-19spring/QC-textbook.pdf

MOOCs

1. https://nptel.ac.in/courses/106106232
2. https://www.coursera.org/lecture/introduction-to-quantum-information/1-quantum-computing-XVPA2

Course Code: MCS-653 **Course Title: (D)Biologically Inspired Computing**

Credit: 4 **LTP: 4-0-0**

Course objectives:

1. To introduce the characteristics of natural agents and building blocks involved in biological processes
2. To provide an understanding on the application of bio inspired algorithms to solve complex problems
3. To provide insights into the implementation of bio inspired algorithms

Course outcomes:
After successful completion of the course, the students will be able to

1. Understand phenomena guiding biological processes through self-organization and adaptability

2. Visualize the effect of low-level interactions on high-level phenomena

3. Analyze complex engineering problems and solve them by adapting biological processes suitably

4. Design and implement simple bio-inspired algorithms

UNIT I

Artificial neural networks: Pattern classification, Single and multilayer perceptrons, back propoagation, pattern association, Hebbian learning, Hopefield networks, Bidirectional associative memory networks, competitive learning, Kohonen's self organizing maps. Deep neural networks, Applications of neural networks

UNIT II

Genetic algorithms - Genetic operators, the building block- schema theory, hypothesis and schema theorem, implicit parallelism. Exploration versus exploitation. Representation, reproduction, crossover and mutation operators. Crossover and mutation rates. Selection mechanisms – Fitness proportionate – ranking and tournament selection, Stochastic models of GAs- reliability model, branching-process model, Markov models.

UNIT-III:

Convergence analysis, Analysis of Selection, Analysis of crossover, Analysis of mutation-crossover versus mutation. Non-canonical Genetic Algorithms: deception, evolution strategies. Evolutionary algorithms, evolutionary programming. genetic programming. Applications of Evolutionary Algorithms in diverse fields - constrained optimization, combinatorial optimization, learning

UNIT IV

Swarm Intelligence: Introduction to Swarm Intelligence, Particle Swarm Optimization (PSO), Anatomy of a particle, Velocity and Position updation, PSO topologies, Control parameters. Ant Colony Optimization (ACO), Pheromone updation and evaporation. Cuckoo Search (CS) algorithm, Bat Search Algorithm (BSA)

UNIT V

Applications of PSO in Job scheduling on computational grid, PSO for data mining. ACO for data mining, solving the traveling salesman problem using ACO, 0/1 knapsack problem, Swarm Based network management, multi-objective optimization, swarm-bots, Applications of Cuckoo Search (CS) algorithm, Bat Search Algorithm (BSA).

Text Books:

1. David E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addision Wesley, MA, 1989.

2. David E. Goldberg: Genetic algorithms in search, optimization and machine learning, Addison Wesley, 1993. Eric Bonabeau, Marco Dorigo and Guy Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999

4. Leandro Nunes De Castro, Fernando Jose Von Zuben, "Recent Developments in Biologically Inspired Computing", Idea Group Publishing, 2005.

5. Laurene Fausett, "Fundamentals of neural networks: architectures, algorithms, and applications", Prentice-Hall, 1994

References:

1. IEEE Transactions on Evolutionary Computation.

2. Evolutionary Computation, MIT Press.

3. Xin-She Yang, "Recent Advances in Swarm Intelligence and Evolutionary Computation", Springer International Publishing, Switzerland, 2015.

Course Code: MCS-654 **Course Title: Dissertation (Research Project Part-II)**

Credit: 8 **LTP: 0-0-16**