Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
Syllabus outline of M. Sc. (Computer Science) First Year
Effective from Academic Year 2019-20

Program Outcomes:

After Successful Completion of M.Sc. (Computer Science) Students will be able to:

1. Find the correct Solution of complex Problem in CSIR/NET/SET/GATE.
2. Contribute to the design of software products and applications.
3. Become a software developer in IT industry.
4. Acquire jobs in Data Science and Machine learning areas.
5. Get placed in teaching / research units.
6. Adopt specialized skills in independently analyze model real world problems and solve it by applying their underlying Knowledge of the subject.

Academic Year 2021-22

Semester	Course Code	Course Name	Marks		Total	
			Internal	End Semester	Marks	Credits
Semester - I	**Core Course (CC) (Compulsory Course)**					
	M. Sc. CS-CC-1	Design and Analysis of Algorithm	40	60	100	4
	M. Sc. CS-CC-2	Computer System Architecture	40	60	100	4
	M. Sc. CS-CC-3	Object Oriented Programming using Python	40	60	100	4
	Discipline Specific Elective (DSE) Course (Choose Any One)					
	M. Sc. CS - DSE-1	Mobile Computing	40	60	100	4
	M. Sc. CS -DSE-2	Data Mining	40	60	100	4
	Practical / Lab Course					
	M. Sc. CS - Lab-1	Lab-Course1 (DAA + CSA)	40	60	100	4
	M. Sc. CS -Lab-2	Lab-Course2 (OOP's using Python+ DSE)	40	60	100	4
	M. Sc. CS -SDC-1	Seminar	25	---	25	1
		Total(I)			625	25
Semester	**Course Code**	**Course Name**	**Marks**		**Total**	
			Internal	**End Semester**	**Marks**	**Credits**
Semester – II	**Core Course (CC) (Compulsory Course)**					
	M.Sc.CS-CC-4	Numerical Methods	40	60	100	4
	M.Sc.CS-CC-5	Compiler Design	40	60	100	4
	M.Sc.CS-CC-6	Web Programming Using ASP.Net	40	60	100	4
	Discipline Specific Elective (DSE) Course (Choose Any One)					
	M.Sc. CS-DSE-3	Internet of Things	40	60	100	4
	M.Sc. CS-DSE-4	Bioinformatics	40	60	100	4
	Practical / Lab Course					
	M. Sc.CS-Lab-3	Lab-Course3 (NM+CD)	40	60	100	4
	M. Sc.CS-Lab-4	Lab-Course4 (ASP.Net+DSE)	40	60	100	4
	M. Sc. CS -SDC-2	Seminar		--	25	1
		Total (II)			625	25
	Total (I + II i.e. Sem I + Sem II)				**1250**	**50**

M. Sc. CS – CC -1

Design and Analysis of Algorithm

Total Teaching hours: 60 **Marks: 100**

Learning Objectives

- The objective of the course is to teach techniques for effective problem solving in computing.
- The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to solve a given problems.

Learning Outcomes:

After completion of this course students should be able to:

- Learn good principles of algorithm design.
- Learn how to analyze algorithms and estimate their time complexity.
- Become familiar with fundamental data structures and their implementation
- Become accustomed to the description of algorithms in both functional and procedural ways.
- Learn how to apply their theoretical knowledge in practice (via the practical component of the course).

Contents	Teaching Hours
UNIT I: Introduction A simple example of design using insertion sort, pseudo code for insertion sort, time complexity. Performance Analysis – Space complexity and Time complexity (posteriori testing and priory approach), Asymptotic Notations (O, Ω, Θ), Examples on Asymptotic Notations, Polynomial vs. Exponential Algorithms. Average, Best and Worst-case complexity.	15
UNIT II: Divide and Conquer Algorithms, Greedy Algorithms Introduction to Divide and Conquer Algorithms, Finding the Maximum and Minimum, Quick sort (Derivation of Average case analysis and Worst-case analysis), Binary Search (Derivation of average case analysis), and Strassen's Matrix Multiplication. Introduction to Greedy Algorithms – Fractional Knapsack problem, Minimum cost spanning trees, Kruskal's and Prim's	15

Algorithms, Optimal Merge patterns and Single-Source Shortest Paths.	
UNIT III: Dynamic Programming, Back Tracking and Branch &Bound Algorithms Dynamic Programming Definition - All-pairs shortest paths, Traveling salesman problem and optimal parameterization for product of sequence of matrices. Back tracking and Branch and Bound Algorithms Introduction – N queens Problem, Sum of Subsets problem using Back tracking algorithms. Traveling Salesman problem using branch and bound method.	15
UNIT IV: Graphs and Heaps & Lower bound Theory Graphs and Heaps Definitions – Adjacency Matrix, Adjacency Lists. Breadth First Search and Traversal, Depth First Search and Traversal. Priority Queues using Heap and Design of Heap sort using. Priority Queues Lower bound Theory A brief introduction to comparison trees and NP hard and NP complete problems.	15

REFERENCE BOOKS:

1. Horowitz, Sahni, Rajasekaran, "Fundamentals of Computer Algorithms" Galgotia Publications, 1996.
2. Donald E. Knuth, "The Art of Computer Programming" Volume 3, Sorting and Searching, Second Edition, Pearson Education.
3. Donald E. Knuth, "The Art of Computer Programming" Volume 1, fundamental algorithms, Third Edition, Pearson Education.
4. Richard F. Gilberg, Behrouz A, Forouzan "Data structures A Pseudocode Approach with C".

M.Sc.CS-CC-2

Computer System Architecture

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- It will cover machine level representation of data, instruction sets, computer arithmetic, CPU structure and functions, memory system organization and architecture, system input/output, multiprocessors, and digital logic.
- The emphasis is on studying and analyzing fundamental issues in architecture design and their impact on performance.
- To acquire the basic knowledge of microprocessor and application of knowledge to understand electronics circuits.
- To prepare students to perform the analysis and design of various combinational and sequential logic circuits.

Learning Outcomes:

After completion of this course students should be able to:

- Analyze and evaluate computer performance with memory and control unit.
- Do the assembly language programming of microprocessor 8086.
- Have a complete understanding of the fundamental concepts and techniques used in microprocessor.
- Understand and examine the structure of various microprocessor and its application in real world.
- Understand, analyze and design various combinational and sequential circuits.

Contents	Teaching Hours
UNIT I: DESIGN METHODOLOGY AND PROCESSORS DESIGN Introduction to system modeling, Combinational and Sequential circuit design, register level design, Processor level components, Queuing models, Simulation, Processor organization, Information representation, Instruction sets, Instruction format & types, RISC, CICS processor concepts.	15

UNIT II: CONTROL UNIT AND MEMORY General model of control unit, Hardwired control unit, Micro-programmed control unit, Memory Characteristics, Memory Hierarchies, Classification chart of memory, Primary and Secondary memories, Virtual and Cache memory, High speed Memories: Interleaved and Associative memory.	15
UNIT III: INTRODUCTION TO MICROPROCESSOR 8085 Introduction to Microprocessor, History of Microprocessor, Buses, Format of instruction, Introduction to 8085 processor with Features, Architecture of Intel 8085, Functional Pin Diagram of Intel 8085, Pin description, Addressing Modes, Instruction set, Programming of 8085.	15
UNIT – IV INTRODUCTION TO OTHER MICROPROCESSOR Features of Intel 8086, Architecture of Intel 8086, Functional Pin Diagram of Intel 8086, Addressing Modes of Intel 8086, Instruction set of Intel 8086, Assembly Language Programming of 8086.	15

REFERENCE BOOKS:

1. Computer Architecture & Organization -J. P. Hayes. (MGH)
2. Microprocessor 8086 by B. Ram
3. 8085 Microprocessor Architecture Programming & Application- Ramesh Gaonkar, Willey Estern.
4. 8086/8088 Microprocessor Family –Liu Gibson (MGH)

M.Sc.CS-CC-3

Object Oriented Programming using Python

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

The learning objectives of this course are:

- To learn how to design and program Python applications.
- 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 define the structure and components of a Python program.
- 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.

Learning Outcomes:

- After completion of this course students should be able to:
- Demonstrate the principles of object-oriented programming and the interplay of algorithms and data structures in well-written modular code;
- Solve problems requiring the writing of well-documented programs in the Python language, including use of the logical constructs of that language;
- Demonstrate significant experience with the Python program development environment.

Contents	Teaching Hours
Unit I: Introduction to Python Installation and Working with Python, Understanding Python variables, Python basic Operators, Understanding python blocks, Declaring and using Numeric data types: int, float, complex, Using string data type and string operations, Defining list and list slicing, Use of Tuple data type, Conditional blocks using if, else and elif, Simple for loops in python, For loop using ranges, string, list and dictionaries, Use of while loops in python, Loop manipulation using pass, continue, break and else, Programming using Python conditional and loops block	15

Unit II: Python Functions, Modules and Packages Organizing python codes using functions, organizing python projects into modules, importing own module as well as external modules, Understanding Packages, Powerful Lamda function in python, Programming using functions, modules and external packages, building blocks of python programs, understanding string in build methods, List manipulation using in build methods, Dictionary manipulation, Programming using string, list and dictionary in build functions.	15
Unit III: Python File Operation and OOPs Reading config files in python, Writing log files in python, Understanding read functions, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Manipulating file pointer using seek, Programming using file operations, Concept of class, object and instances, Constructor, class attributes and destructors, Real time use of class in live projects inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes, Programming using OOPs support.	15
UNIT IV: Python Regular Expression, Exception Handling &Database Interaction Powerful pattern matching and searching, Power of pattern searching using regex in python, Real time parsing of networking or system data using regex, Password, email, url validation using regular expression, Pattern finding programs using regular expression, Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with error code, Programming using Exception handling, SQL Database connection using python, Creating and searching tables, Reading and storing config information on database, Programming using database connections. NumPy: Arrays, Array indexing Datatypes, Array math, Broad casting, NumPy Documentation.	15

REFERENCE BOOKS:

1. Starting Out with Python plus My Programming Lab- Tony Gaddis, Pearson
2. Learning Python- Mark Lutz, O'Reilly, 5th edition.

M.Sc.CS-DSE-1
Mobile Computing

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- To impart fundamental concepts in the area of mobile computing,
- To provide a computer systems perspective on the converging areas of wireless networking, embedded systems, and software.
- To introduce selected topics of current research interest in the field.

Learning Outcomes:

- After completion of this course students should be able to:
- Determine solutions using problem solving principles, logic and systematic methodologies.
- Evaluate the architecture and principles of operation of computer systems and networks.
- Synthesize principles and theories of computer science and software engineering for application to different computing paradigms.

Contents	Teaching Hours
Unit I: Introduction to GSM Application, A Short History of Wireless Communication, A Market for Mobile Communication, Some Open Research Topic, A Simplified reference Model, Basic Cellular System, Performance Criteria, Operation of Cellular System, Planning a Cellular System, Analog Cellular System.	15
Unit II: Medium Access Control (MAC) Motivation for specialized MAC, SDMA, FDMA, TDMA, CDMA	15
Unit III: Telecommunication System and GSM Mobile services, System architecture, Radio interface, DECT, System architecture, Protocol architecture, TETRA, UMTS and IMT-2000	15
Unit IV: Wireless LAN Infrared Vs radio transmission, Infrastructure and along Network, IEEE 802.11, HIPERLAN, Bluetooth	15

Reference Books:

1. Mobile Communications Second Edition – By Jochen Schiller (Pearson Education)
2. Mobile Cellular Telecommunications Second Edition-By William, C.Y.Lee (Mc-Graw-Hill)

M.Sc.CS-DSE-2
Data Mining

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- Understanding of the value of data mining in solving real-world problems.
- Understanding of foundational concepts underlying data mining.
- Understanding of algorithms commonly used in data mining tools.
- Ability to apply data mining tools to real-world problems.

Learning Outcomes:

- After completion of this course students should be able to:
- Evaluate models/algorithms with respect to their accuracy.
- Demonstrate capacity to perform a self-directed piece of practical work that requires the application of data mining techniques.
- Critique the results of a data mining exercise.
- Develop hypotheses based on the analysis of the results obtained and test them.

Contents	Teaching Hours
Unit I Introduction to Data mining with related concepts Introduction to Data mining with related concepts: Basic Data Mining Tasks, Data Mining Issues. Knowledge Discovery in Databases (KDD Process). OLTP system, Information Retrieval system, Decision Support Systems, Multidimensional Schemas, OLAP, Web Search Engines.	15
Unit II: Data Mining Techniques: Classification : Introduction to Data Mining Techniques. A statistical Perspective on Data Mining, Decision Trees, Neural Networks. Issues in Classification, Bayesian Classification, and Distance Based Algorithms, Decision Tree Based Algorithm: CART, Neural Network-Based Algorithm: NN Supervised Learning.	15
Unit III: Clustering and Association Rules Clustering and Association Rules, Introduction to Clustering, Outliers, K-Means clustering, Nearest Neighbor Algorithm, BIRCH algorithm. Introduction to Association Rules, Large Item sets, Basic Algorithms: Apriori Algorithm, Data Parallelism, Comparing Approaches.	15

Unit IV: Applications and Trends in Data Mining Data Mining Applications: Web mining, Image mining, Text mining, Spatial mining, Fraud Detection, CRM (Customer Relationship Management), Education, Health Care etc., Data Mining System Products.	15

Reference Books:

1. Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Morgan Kauf Mann Publishers.

2. Margaret H. Dunham, S. Sridhar, Data Mining – Introductory and Advanced Topics, Pearson Education

M. Sc. CS -Lab-1
(Design Analysis and Algorithm)

Learning Objectives:

- Reinforce basic design concepts (e.g., pseudocode, specifications, top-down design)
- Knowledge of algorithm design strategies
- Familiarity with important algorithms
- Ability to analyze time and space complexity

Learning Outcomes:

- After completion of this course students should be able to:
- write programs in MATLAB to solve problems using algorithm design techniques such as Divide and Conquer, Greedy, Dynamic programming, and Backtracking.
- implement Quick sort, Merge sort algorithm, BFS and DFS algorithms.
- implement backtracking algorithm for the N-queens problem.

LIST OF PRACTICALS

1. Program in MATLAB to find Maximum Number.
2. Program in MATLAB to find Minimum Number.
3. Program in MATLAB to check Given number is Even or Odd.
4. Program in MATLAB Sort numbers in ascending order.
5. Program in MATLAB Sort numbers in descending order.
6. Program in MATLAB to implement iterative binary search.
7. Program in MATLAB to implement linear search.
8. Program in MATLAB for matrix addition.
9. Program in MATLAB to multiply two matrices.
10. Program in MATLAB to multiply matrix using Strassen Matrix Multiplication.
11. Program in MATLAB to perform insertion sort.
12. Program in MATLAB to calculate exponential Xn.
13. Program in MATLAB for calculating average of numbers.
14. Program in MATLAB for calculating sum of even numbers.
15. Program in MATLAB for calculating sum of odd numbers.

M. Sc. CS Lab-1
Computer System Architecture

Learning Objectives:

- It will cover machine level representation of data, instruction sets, computer arithmetic, CPU structure and functions, memory system organization and architecture, system input/output, multiprocessors, and digital logic.
- The emphasis is on studying and analyzing fundamental issues in architecture design and their impact on performance
- To acquire the basic knowledge of microprocessor and application of knowledge to understand electronics circuits.
- To prepare students to perform the analysis and design of various combinational and sequential logic circuits.
- To impart to you the concepts of sequential circuits enabling you to analyze sequential systems in terms of state machines.

Learning Outcomes:

After completion of this course students should be able to:

- Analyze and evaluate computer performance with memory and control unit concerned.do the assembly language programming of microprocessor 8086.
- Have a complete understanding of the fundamental concepts and techniques used in microprocessor.
- Understand and examine the structure of various microprocessor and its application in real world.
- Analyze and design various combinational and sequential circuits. understand the basics of computer hardware and how software interacts with computer hardware.

List of Practical:

1. To perform and verify the truth tables of basic gates and derived gates.
2. To perform and verify the truth table of half adder.
3. To perform and verify the truth table of half Subtractor.
4. To perform and verify the truth table of multiplexer.
5. To perform and verify the truth table of De-multiplexer.
6. To perform and verify the truth table of encoder.
7. To perform and verify the truth table of Decoder.
8. Assembly language Programs of 8085.
9. Write an ALP for checking various operations of 8086.
10. Write an ALP for addition and subtraction for 8086.

11. Write an ALP for multiplication and division for 8086.
12. Write an ALP for multiplication and division for 8086.
13. Write an ALP to find largest number from array for 8086.
14. Write an ALP to find smallest number from array for 8086.

M. Sc. CS -Lab-2
OOP's using Python

Learning Objectives:

- To acquire programming skills in core Python.
- To acquire Object Oriented Skills in Python.
- To develop the skill of designing Graphical user Interfaces in Python.

Learning Outcomes:

After completion of this course students should be able to:

- Explain basic principles of Python programming language.
- Implement object-oriented concepts.
- Implement database and GUI applications.

LIST OF PRACTICALS

1) Program to demonstrate Constant variable.
2) Program to demonstrate scope of a variable.
3) Program to demonstrate branching statement.
4) Program to demonstrate Looping statement.
5) Program to demonstrate simple class.
6) Program to demonstrate String class and its method.
7) Program to demonstrate String Buffer and its method.
8) Program to demonstrate inheritance and its Types.
9) Program to demonstrate package.
10) Program to demonstrate polymorphism.
11) Program to demonstrate database connectivity.
12) Program to demonstrate networking**.**

M. Sc. CS -Lab-2
DSE1(Mobile Computing)

Learning Objectives

- Understand the fundamentals of wireless networks.
- Learn and analyze the different wireless technologies.
- Evaluate Ad-hoc networks and wireless sensor networks.
- Understand and evaluate emerging wireless technologies

Learning Outcomes:

After completion of this course students should be able to:

- Explain the basic concepts of wireless network and wireless generations.
- Demonstrate the different wireless technologies such as CDMA, GSM, GPRS etc
- Appraise the importance of Adhoc networks such as MANET and VANET
- Understand design considerations for wireless networks

List of Practical:

1 To implement Code Division Multiple Access (CDMA).
2 To study frequency reuse concept.
3 To study basic concept of J2ME.
4 To study various classes (such as TextBox, ChoiceGroup, Drop Down menus etc.) and their implementation in J2ME.
5 To design a simple WML page using various WML tags.
6 To implement mobile network using NS2.
7 Study Assignment 1: Detailed study of Bluetooth
8 Study Assignment 2: Detailed study of Wireless Application Protocol.
9 Set up and configuration of access point
10 Study Assignment 3: To study network security software.

M. Sc. CS -Lab-2
DSE2(Data Mining)

Learning Objectives:

- To introduce basic concepts and techniques of Data Mining.
- To develop skills of using recent data mining software for solving practical problems.
- To study complex data types for web mining.

Learning Outcomes

- After completion of this course students should be able to:
- Understand Data Mining Principles
- Identify appropriate data mining algorithms to solve real world problems.
- Compare and evaluate different data mining techniques like classification, prediction, clustering and association rule mining.
- Describe complex data types with respect to spatial and web mining.

List of Practical:

1.Demonstration of Orange and tools for data mining.
2. Demonstration of preprocessing technique on the existing data set of Orange
a. Select weather. nominal data set and preprocess the .ARFF file.
b. Select weather. numeric data set and preprocess the .ARFF file.
3. Demonstration of classification technique using J48 algorithm on
a. Select weather. nominal dataset
b. Select weather. numeric dataset.
c. Create your own .ARFF file and classify it and visualize the classifier error.
4. Demonstration of classification technique using id3 and naive bayes algorithm on
a. Select weather. nominal dataset
b. Select weather. numeric dataset.
c. Create your own .ARFF file and classify it and visualize the classifier error.
5. Demonstration of Association Rule using Apriori algorithm
a. Dataset test. ARFF using Apriori algorithm.
b. Dataset contact lenses. ARFF using Apriori algorithm
6. Demonstration of clustering rule process on iris dataset. ARFF using simple k-means algorithm. (what is the meaning of underline sentence?)
7. Demonstration of clustering rule process on own dataset student.arff using simple k-means algorithms.

8. Demonstration of Predicting the price of house using linear and nonlinear regression algorithm based on different parameters of house.
(Apart from the above list, concerned teacher can conduct different kinds of programs for the better understanding Subjects)

M. Sc. FY Semester II
M. Sc. CS-CC-4
Numerical Methods

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- Understand the fundamental principles of digital computing, including number representation and arithmetic operations.
- Understand the linkage between accuracy, stability and convergence.
- Perform error analysis for arithmetic operations.
- Understand the propagation of errors through complex numerical algorithms.

Learning Outcomes:

- After completion of course students should be able to-
- Develop and implement numerically stable and accurate algorithms for all the basic tasks of computational science and engineering:
- Develop stable algorithms for solving linear systems of equations.
- Develop efficient and stable algorithms for finding roots of non-linear equations.
- Implement numerically stable recursion algorithms for evaluating mathematical functions.
- Understand the use of interpolation for numerical differentiation and integration.

Contents	Teaching Hours
Unit I Computer Arithmetic & Solution of Algebraic equations Computer Arithmetic: Floating Point representation of Numbers, Arithmetic operation with Normalized floating point, Solution of algebraic equations: Bisection method, Method of false position, Newton-Raphson Method, Secant Method.	15
Unit II Interpolation and Numerical Differentiation and Numerical Integration Finite differences [forward & backward], Lagrange interpolation, Difference tables, Numerical differentiation & numerical integration, Trapezoidal rule, Simpson's 1/3 Rule, Simpson's 3/8 Rule.	15

Unit III Matrices and linear system of equations Introduction, Solution of linear system, Matrix inversion method, Gaussian elimination method, Modification of gauss method to compute the inverse.	15
Unit IV Curve Fitting Least square Curve fitting, Fitting a straight line, Nonlinear curve fitting polynomial of nth degree.	15

Reference Books:

1. Computer Orient Numerical Analysis by V. Rajaraman
2. Numerical Analysis by Sastri

M. Sc. CS CC-5
Compiler Design

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- To understand basic concepts and designing of assembler, Macro processor and role of static and dynamic loaders and linkers.
- To understand the need to follow the syntax in writing an application program and
- To learn the how the analysis phase of compiler is designed to understand the programmer 's requirements without ambiguity.
- To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time.

Learning Outcomes:

After completion of this course students should be able to-

- Identify the relevance of different system programs.
- Describe the various data structures and passes of assembler design.
- Implement different parsers for given context free grammars.
- Synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time.

Contents	Teaching Hours
UNIT I: Introduction to Compilers and Programming Languages Compilers and translators, the structure of compiler, Compiler writing tools, Definition of P.L., High level Programming Languages., Lexical and syntactic structure of a language, Data structures, Operators, Statements. Lexical Analysis: Introduction to Lexical analysis, Role of a Lexical analyzer, A simple approach to the design of lexical analyzer, Regular expressions.	15
UNIT II: Syntax Analysis and Basic Parsing Techniques Finite automata, minimizing number of states of a DFA, Implementation of a lexical analyzer. Context free grammars, Introduction to parsers, Shift reduce parsing, Top-down parsing, Operator Precedence parsing, Predictive parsers.	15

UNIT III: Syntax Directed Translation and symbol table Introduction to Syntax directed Schemes, Implementation of Syntax directed translators, Intermediate code, Postfix notation and evaluation of postfix expressions, Parse trees and syntax trees, the contents of a symbol table, Data structures for a symbol table.	15
UNIT IV: Error detection and recovery, Introduction to Code Optimization Introduction to Errors, Lexical phase errors, Syntactic phase errors, Semantic errors, Sources of optimization, Loop optimization.	15

Reference Books:

1. Principals of Compiler Design by Alfred V. Aho, Jeffrey D. Ullman
2. Compilers Principles, Techniques and Tools By A.V. Aho, R. Shethi and J.D.
3. Introduction to system software By D. M. Dhamdhere.

M.Sc.CS-CC-6

Web Programming using Asp.net

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- Asp.net Helps students to create their own web applications.
- According to Market every site is divided into two categories static or dynamic and subject like asp.net helps to design and develop static as well as dynamic websites.
- Asp.net technology is one of good choice to develop major project of final year PG students.

Learning Outcomes:

After completion of course students should be able to-

- Create a Web form with server controls.
- Separate page code from content by using code-behind pages, page controls, and components.
- Display dynamic data from a data source by using Microsoft ADO.NET and data binding.
- Debug ASP.NET pages by using trace.

Contents	Teaching Hours
Unit-I: Introduction to Web Technology & Asp.net Web Support Languages, Types of .net Application: web, Desktop, Mobile. Role of Web Browser and Web Server Understanding HTTP, LAN, WAN, Client side and Server-side Scripting, Introduction to Ajax and WCF, Introduction to .net, .net Framework Installing visual studio.net, Asp Vs Asp.net, Asp.net Web page life cycle, Asp.net Web form, Page Directive	15
Unit-II: Asp.net Applications, CSS and Themes Creating Asp.net Web Application, Auto Post back property, HTML controls Vs Web controls Code Window& Design Window, Server-Side controls, Exception Handling, what is CSS? Types of CSS, Theme, Name skin within a Theme.	15
Unit-III: Redirecting User to Another Page & Master Pages. Redirecting Options Response, Redirect Server. Transfer Cross Page Post back, Passing Values between pages, Introduction to Master Page, Content Place Holder and Content tags, Accessing	15

Controls of Master page in Content page, Master page with Menus.	
Unit-IV: User Controls, Validation, State Management and Web Services Creating User Control, Required Field Validator, Compare Validator, Range Validator, Regular Expression Validator, Custom Validator, Query String, View State, Hidden Field, Cookies, Session, Creating Web Services, Web Methods, Database Oriented Asp.net ,Web Application with Grid View.	15

Reference Books:

1. Asp.net 4.0 Black Book
2. Mastering Asp.net, BPB Publication, Russel.
3. Asp.net the Complete Reference: Matthew Macdonald

M. Sc. CS-DSE-3
Internet of Things

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- The objective of the course is to
- Vision and Introduction to IoT.
- Understand IoT Market perspective.
- Data and Knowledge Management and use of Devices in IoT Technology.
- Understand State of the Art – IoT Architecture.
- Real World IoT Design Constraints, Industrial Automation and Commercial Building Automation in IoT.

Learning Outcomes

After completion of this course students should be able to-

- Design some IOT based prototypes.
- Implement Algorithms for IoT based Applications

Contents	Teaching Hours
Unit-I: Introduction and concepts Definition and characteristics of IoT. Physical Design of IoT, Things in IoT, IoT Protocols. Logical Design of IoT- IoT functional blocks, IoT communication models. IoT enabling Technologies-Wireless sensor networks, cloud computing, big data analytics, communication protocols, embedded systems. IoT Levels and deployment templates-IoT Level1 to IoT Level6.	15
Unit-II: Domain Specific IoTs Introduction Home automation- Smart lighting, smart appliances, intrusion detection, smoke or gas detectors. Cities-Smart parking, smart lighting, smart roads, structural help monitoring, surveillance, emergency response. Environment-Weather monitoring, Air pollution monitoring, forest fire detection, river flood detection. Retail- Inventory management, smart payments, smart vending machines	15

Logistics- Route generation and scheduling, fleet tracking, ship monitoring, remote vehicle diagnostic. Agriculture- smart irrigation, green house control. Industry- machine diagnostic, prognosis, indoor air quality monitoring. Health and Lifestyle	
Unit-III: IoT vs M2M and Developing IoTs M2M, Difference between IoT and M2M, Difference between SDN and NFV for IoT- software defined networking and network function virtualization, IoT Code generator.	15
Unit-IV: IoT Design Methodology Purpose and requirement specification, Process specification, Domain model specification, Information model specification, Service specification, IoT level specification, Functional View specification, Operational View specification, Device and component integration, Application Development with Python.	15

Reference Books:

1. Internet of Things (A hands on approach) By: Vijay Madisetti and Arsh deep Bagha
2. Rethinking the Internet of Things: A scalable approach to connecting everything By Francis DaCosta.
3. Designing the Internet of Things by Adrian McEwen & Hakim Cassimally.

M. Sc. CS-DSE-4
Bioinformatics

Total Teaching Hours: 60 **Total Marks: 100**

Learning Objectives:

- To improve the programming skills of the student
- To let the students, know the recent evolution in biological science.

Learning Outcomes:

- Upon completion of this course, students should be able to:
- Develop bioinformatics tools with programming skills.
- Apply computational based solutions for biological perspectives.
- Pursue higher education in this field.
- Practice life-long learning of applied biological science.

Contents	Teaching Hours
UNIT: I Introduction Introduction to Bioinformatics and Computational Biology, Biological sequences, Biological databases, Genome specific databases, Data file formats, Data life cycle, Database management system models.	10
UNIT: II Algorithms in Bioinformatics Sequence Analysis, pairwise alignment, Dynamic programming algorithms for computing edit distance, string similarity, shotgun DNA sequencing, end space free alignment. Multiple sequence alignment, Algorithms for Multiple sequence alignment, Generating motifs and profiles, Local and Global alignment, Needleman and Wunsch algorithm, Smith Waterman algorithm, BLAST, PSIBLAST and PHIBLAST algorithms.	20
UNIT: III Introduction to Phylogenetics Introduction to phylogenetics, Distance based trees UPGMA trees, Molecular clock theory, Ultra metric trees, Parsimonious trees, Neigh bour joining trees, trees based on morphological traits, Bootstrapping. Protein Secondary structure and tertiary structure prediction methods, Homology modeling, abinitio approaches, Threading, Critical Assessment of Structure Prediction, and Structural genomics	15

UNIT: IV Neural Networks Applications in Bioinformatics Sequence Encoding and Output Interpretation, Sequence Correlations and Neural Networks, Prediction of Protein Secondary Structure, Prediction of Signal Peptides and Their Cleavage Sites, Applications for DNA and RNA Nucleotide Sequences, Prediction Performance Evaluation, Different Performance Measures.	15

Reference Books:

1. Lesk, A. K., "Introduction to Bioinformatics" 4th Edition, Oxford University Press, 2013
2. Baldi, P. and Brunak, S., "Bioinformatics: The Machine Learning Approach" 2nd Edition, MIT Press, 2001.
3. Dan Gusfield, "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology" Cambridge University Press, 1997.
4. Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., "Biological Sequence Analysis

M. Sc. CS-Lab-3(Numerical Methods)
Lab-Course3

Learning Objectives

- To Understand Accuracy and precision with examples.
- To locating roots of the equations using Graphical method, Bisection method and false Position method. Understand one-point iterative method to find True roots.
- To Know the Open method like Newton Rapson's Method
- To fit curve into straight line.
- To find inverse matrix inversion method.

Learning Outcomes:

- Learned the difference between Accuracy and Precision and types of errors.
- Finding roots using Graphical method, Bisection method and False position method.
- Solve a iteration method to obtained true roots.
- Evaluate the True roots using Open method: Newton's Rapson method, Newton Rapson method.
- To understand Gauss Elimination Method.

List of Practical:

1. Program in PYTHON for error calculation.
2. Program in PYTHON for successive bisection method.
3. Program in PYTHON for Newton Raphson method.
4. Program in PYTHON for false position method.
5. Program in PYTHON for secant method.
6. Programs in PYTHON for first order interpolation formula.
7. Programs in PYTHON for second order interpolation formula.
8. Programs in PYTHON for third order interpolation formula.
9. Program in PYTHON to find determinant of matrix.
10. Program in PYTHON to find inverse of matrix.
11. Programs in PYTHON to implement forward difference table.
12. Programs in PYTHON to implement backward difference table.

M. Sc. CS Lab-3 (Compiler Design)
Lab-Course3

Learning Objectives:

- To understand the need to follow the syntax in writing an application program and to learn the how the analysis phase of compiler is designed to understand the programmer ‘s requirements without ambiguity.
- To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time.

Learning Outcomes

After completion of course students should be able to-

- Identify the relevance of different system programs.
- Describe the various data structures and passes of assembler design.
- Identify the need for different features and designing of macros.
- Construct different parsers for given context free grammars. compiler is designed to understand the programmer ‘s requirements without ambiguity.
- To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time

List of practical:

1. Write a C program to identify whether a given line is a comment or not
2. Write a C program to recognize strings under 'a', 'a*b+', 'abb'.
3. Write a C program to test whether a given identifier is valid or not.
4. Write a C program to simulate lexical analyzer for validating operators
5. To Study about Lexical Analyzer Generator (LEX) and Flex (Fast Lexical Analyzer)
6. Implement following programs using Lex. a. Create a Lexer to take input from text file and count no of characters, no. of lines & no. of words.
7. Write a Lex program to count number of vowels and consonants in a given input string.
8. Implement following programs using Lex.
9. Write a Lex program to print out all numbers from the given file.
10. Write a Lex program to count the number of comment lines in a given C program. Also eliminate them and copy that program into separate file.
11. Write a C program for implementing the functionalities of predictive parser for the mini language.
12. Write a C program for constructing recursive descent parsing