Rajarshi Shahu Mahavidyalaya (Autonomous), Latur

Syllabus
(2022-23)

Under CBCS

Two Year Post Graduate Program in

Computer Science

Department of Computer Science

M. Sc. (Computer Science)

Syllabi Approved by the Board of Studies in Computer Science with effect from June, 2022

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Design and Analysis of Algorithm

Course Code: P-DAA-126 **Credits: 4**

Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To analyze performance of algorithms.
- To choose the appropriate data structure and algorithm design method for a specified application.
- To solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.
- To understand & implement tractable and intractable problems.

Course Outcomes: After completion of this course students will be able to:

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

Unit No.	Contents	No. of Lect.
Unit I Introduction & Overview of Data Structure	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. Arrays, Linked List, Stack, Queue, Trees & Graphs	**15 Lectures**
UNIT II: Divide and Conquer Algorithms, Greedy Algorithms	Introduction to Divide and Conquer Algorithms, Binary Search, Finding the Maximum and Minimum, Merge- Sort, Quick sort, Strassen's Matrix Multiplication. Introduction to Greedy Algorithms – Fractional Knapsack problem, Minimum cost spanning trees, Kruskal's Algorithm and Prim's Algorithm, Optimal Merge Patterns, Single-Source Shortest Paths.	**15 Lectures**
UNIT III: Dynamic Programming,	Dynamic Programming Definition – Multistage Graphs, All-pairs shortest paths, Single-Source Shortest Paths, Optimal Binary search Trees, Traveling salesman problem	**15 Lectures**

Back Tracking and Branch &Bound Algorithms	Back tracking and Branch and Bound Algorithms Introduction: 8-Queens Problem, Sum of Subsets problem using Back tracking algorithms. Traveling Salesman problem.	
UNIT IV: Lower bound Theory	Lower bound Theory – A brief introduction to comparison trees, Ordered Searching, Sorting & Selections. Lower bounds through reductions, techniques for algebraic problems. Introduction to NP-hard and NP-Complete Problems	**15 Lectures**

REFERENCE BOOKS:

1. Horowitz, Sahni, "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. Richard F. Gilberg, Behrouz A, Forouzan "Data structures A Pseudocode Approach with C".

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Computer System Architecture

Course Code: P-CSA-127 **Credits: 4**
Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To understand the structure, function of computer systems
- 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 to understand electronics circuits.
- To identify the elements of instruction sets and their impact on processor Design
- To perform the analysis and design of various combinational and sequential logic circuits

Course Outcomes: After completion of this course students will be able to

- Explain the basics of architectural issues of a digital computer and classify and compute the performance of machines, Machine Instructions.
- Analyze the performance of various classes of Memories, build large memories using small memories for better performance and analyze arithmetic for ALU implementation
- Understand and examine the structure of various microprocessor and its application in real world.
- Understand the basics of hardwired and micro-programmed control of the CPU
- Develop assembly language programs using various programming tools

Unit No.	Contents	No. of Lect.
Unit I : Design Methodology and Processor Design	Introduction to Design Methodology, The Gate Level- Combinational circuits, Sequential circuits, The Register level - Register level components, design methods The Processor level - components, design techniques Queuing models. Introduction to Processor Design, Instruction sets, Fixed Point Arithmetic, ALU Design.	**15 Lectures**
Unit II: Control Design and Memory Organization	General model of control unit, Hardwired control unit, Micro-programmed control unit, Micro-programmed Computers. Memory Technology - Memory Device Characteristics, Random Access Memories, Serial Access Memories. Virtual Memory - Memory Hierarchies, Main Memory Allocation, Segments, Pages & Files High Speed Memories- Interleaved memories, Caches and Associative memories.	**15 Lectures**

Unit III: Introduction to Microprocessor 8085	Introduction to Microprocessor 8085, Instruction Cycle, Timing Diagram, RISC and CICS processors. **Instruction set**- Instruction and data formats, addressing modes, Intel 8085 Instructions, Assembly Language Programming	**15 Lectures**
Unit IV: Introduction to Other Microprocessors & Peripheral Devices	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, Introduction to 80386 Microprocessor, Features of 80836, Architecture of 80836. Address space partitioning, memory & I/O Interfacing, Interfacing Devices & I/O Devices.	**15 Lectures**

REFERENCE BOOKS:

1. Computer Architecture & Organization -John. P. Hayes (MGH II Edition)
2. Fundamentals of Microprocessors & Microcomputers - Badri Ram (Dhanpat Rai Publications (P) Ltd. Fourth Revised & Enlarged Edition)

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Object Oriented Programming Using Python

Course Code: P-OUP-128 **Credits: 4**

Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To understand how to write program in Python.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to build and package Python modules for reusability.
- To learn how to design object-oriented programs with Python classes.
- To learn how to read and write files in Python.
- To implement inheritance, exception handling & RegEx in Python.
- To learn create GUI application using Python

Course Outcomes: After completion of this course students will be able to

- Implement programming skills in core Python.
- Apply Object Oriented Skills in Python
- Design & develop application with Graphical User Interfaces in Python
- Use the ability to handle exceptions, write regular expression & database applications in Python

Unit No.	Contents	No. of Lect.
Unit I: Introduction& Working with Python	Installation and Working with Python, Understanding Python variables, Python Shell, Python IDLE, Different Python IDEs (VS-Code, Py-charm, Sub-lime text etc) Python Data types, Python Operators, Python blocks Control & looping Statements Built in methods/functions on: String, List, Tuples, Dictionary Functions, Modules in Python, organizing Python projects into modules, importing own module as well as external modules, Understanding Packages, Powerful Lambda function in Python, Modules: The from … import statement, A module's name, Making your own modules, The dir function, packages. Brief Tour of the Standard Library: math, date time, turtle, NumPy, SciPy, Panda.	**15 Lectures**
Unit II: OOPs Concept, & File Handling in Python	Concept of class-object/instance, Constructor & its types, Destructor in Python, Types of Variables: Instance Variable, Class Variables, Static Variables, Types of Methods: Instance Method, Class Method, Static Method, Passing Members of One class to Another Class, inner class.	**15 Lectures**

	Inheritance: Single, Multiple, Multilevel, Hierarchical, Hybrid, Python Constructor in Inheritance, Constructor/ Method Overriding, Method Overloading, Operator Overloading Abstract Classes & Interfaces: Abstract Method & Class, Interfaces in Python, Abstract Class Vs Interface File Handling in Python: Types of Files, Reading, Writing Files using Python, With Statement. Understanding read functions, read (), readline() and readlines(), Understanding write functions, write() and writelines(), Manipulating file pointer using seek, Programming using file operations.	
Unit III Graphical User Interface in Python	GUI in Python, The Root Window, Fonts and Colors, Working with Containers, Canvas, Frame, Widgets, Button Widgets, arranging widgets in the Frame, Label, Widget, Message Widget, Text Widget, Scrollbar Widget, CheckButton Widget, RadioButton Widget, Entry Widget, Spinbox Widget, ListBox widget, Menu Widget, Creating Tables	**15 Lectures**
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, Exception Handling: Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with error code, Programming using Exception handling, Database Connection: SQL Database connection using Python, Creating and searching tables, Reading and storing config information on database, Programming using database connections.	**15 Lectures**

REFERENCE BOOKS:

1. Core Python Programming- Dr. R Nageswara Rao (Dreamtech Press)
2. Learning Python- Mark Lutz, O'Reilly, 5th edition.
3. Starting Out with Python plus My Programming Lab- Tony Gaddis, Pearson

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Mobile Application Development

Course Code: P-MAD-129 **Credits: 4**
Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To learn Kotlin Programming Language
- To use Kotlin Programming Language for the development of Android Applications
- To Learn Kuddo & Nutrilicious App for development
- Use of DSLs for complex Application Development

Course Outcomes: After completion of this course students will be able to

- Describe the basic concepts and principles to develop the mobile application
- Develop the Android Applications using Kotlin Programming Language.
- Develop Kuddo & Nutrlicious App using Android Studio
- Develop the complex Application Development

Unit No.	Contents	No. of Lect.
Unit I Learning & Diving into Kotlin	Introducing Kotlin, Diving into Kotlin- Variables & datatypes, Conditional Code, Loops & Ranges, Functions, Null Safety m Equality Checks, Exception Handling, Purpose of Functional programming, Functions, Working with Collections, Scoping Functions, Lazy Sequences.	**15 Lectures**
Unit II Object Orientation in Kotlin & Interoperability with JAVA	Classes & Object Instantiation, Properties, Method, Primary & Secondary Constructors, Inheritance & Overriding rules, Type Checking & Casting, Visibilities, Data classes, Enumerations, Objects & Companions, Generics. Interoperability with JAVA – Using JAVA Code from Kotlin, Using Kotlin Code from JAVA, Best Practices for Interoperability.	**15 Lectures**
Unit III Android App Development with Kotlin: Kudoo App, Nutrilicious	Setting up Kotlin for Android, using Kotlin in Android Studio, Autogenerated Gradle Configuration, adapting your Gradle Configuration using Annotation Processors, Converting java code to Kotlin. Kuddo, a To Do List App: Creating the project, Adding the Recycler View, adding a Room Data Base, using a View Model, Integrating Livedata, Adding New To-Do Items, Enabling Checking off To Do items Nutrilicious: Setting up the Project, Adding RecyclerView to the home screen, Fetching Data from the USDA nutrition API, Mapping JSON Data to Domain Classes, Introducing ViewModel for Search, letting users search Foods, Store user's favorite foods in room database,	**20 Lectures**
UNIT IV	Introducing DSLs, creating a DSL in Kotlin, DSL for Android Layouts with Anko, DSL for Gradle Build Scripts	**10 Lectures**

Kotlin DSLs and Migrating to Kotlin	On Software Migrations, Leading the Change, Partial or Full Migration, Where to start, Tool Support.	

REFERENCE BOOKS:

1. **“**Kotlin for Android App Development”, Peter Sommerhoff , Pearson Publication

2. Android Programming with Kotlin for Beginners: Build Android apps starting from zero programming experience with the new Kotlin programming language – John Horton (Packt)

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Design and analysis of algorithm

Course Code: P-LAC-130 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To analyze performance of algorithms.
- To choose the appropriate data structure and algorithm design method for a specified application.
- To solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.
- To understand & implement tractable and intractable problems.

Course Outcomes: After completion of this course students will be able to:

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

List of Practical

1. Program in Python to implement Quick sort algorithm for sorting a list of integers in ascending order
2. Program in Python to implement Merge sort algorithm for sorting a list of integers in ascending order.
3. Program in Python to implement Bubble Sort algorithm
4. Program in Python to implement sorting of numbers in ascending and Descending order
5. Program in Python to implement greedy algorithm for job sequencing with deadlines
6. Program in Python to implement the DFS algorithm for a graph.
7. Program in Python to implement the BFA algorithm for a graph.
8. Program in Python to implement Binary Search with Divide and conquer approach.
9. Program in Python to multiply matrix using Strassen's matrix multiplication.
10. Program in Python to implement backtracking algorithm for the 8-queens problem.
11. Program in Python to implement the backtracking algorithm for the sum of subsets problem.
12. Program in Python to implements Prim's algorithm to generate minimum cost spanning tree.
13. Program in Python to Kruskal's algorithm to generate minimum cost spanning tree
14. Program in Python to implement Dynamic Programming algorithm for the 0/1 Knapsack problem.

15. Program in Python to implement Dynamic Programming algorithm for the Optimal Binary Search Tree Problem

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Computer System Architecture

Course Code: P-LAC-131 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To understand the structure, function of computer systems
- 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 to understand electronics circuits.
- To identify the elements of instruction sets and their impact on processor Design
- To perform the analysis and design of various combinational and sequential logic circuits

Course Outcomes: After completion of this course students will be able to

- Explain the basics of architectural issues of a digital computer and classify and compute the performance of machines, Machine Instructions.
- Analyze the performance of various classes of Memories, build large memories using small memories for better performance and analyze arithmetic for ALU implementation
- Understand and examine the structure of various microprocessor and its application in real world.
- Understand the basics of hardwired and micro-programmed control of the CPU
- Develop assembly language programs using various programming tools

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 and half subtractor
3. To perform and verify the truth table of multiplexer
4. To perform and verify the truth table of demultiplexer
5. To perform and verify the truth table of encoder
6. To perform and verify the truth table of decoder
7. Write an ALP for addition & subtraction of two 8-bit ,16-bit numbers
8. Write an ALP for multiplication & division of two 8-bit ,16-bit numbers
9. Write an ALP to find smallest & largest number from array for 8086
10. Write an ALP to find square from lookup table
11. Write an ALP to find one's complement & two's complement of 8-bit number & 16- bit number.
12. Write an ALP to shift an 8-bit number left by one bit.
13. Write an ALP to shift an 8-bit number left by two bits.
14. Write an ALP to find square root of a number.
15. Write an ALP to find multibyte addition.

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Object Oriented Programming Using Python

Course Code: P-LAC-132 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To understand how to write program in Python.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to build and package Python modules for reusability.
- To learn how to design object-oriented programs with Python classes.
- To learn how to read and write files in Python.
- To implement inheritance, exception handling & RegEx in Python.
- To learn create GUI application using Python

Course Outcomes: After completion of this course students will be able to

- Implement programming skills in core Python.
- Apply Object Oriented Skills in Python
- Design & develop application with Graphical User Interfaces in Python
- Use the ability to handle exceptions, write regular expression & database applications in Python

Practical List

1. Study of Datatypes in Python
2. Study of Control /Looping statements
3. Study of Functions in Python
    a. Variables-Local Global
    b. Type of Arguments
4. Study of Modules & Packages
    a. Create & Rename Modules
    b. Create Package
5. Study of Lambda Function
6. Study of List & Dictionary Manipulation
    a. List Methods
    b. Dictionary Methods
7. Study of Constructor, Destructor in Python
    a. Simple Class & Object
    b. Constructor & its types
    c. Destructor
8. Study of types of variables & methods in class
9. Study of Inheritance & its types
    a. Single
    b. Multiple
    c. Multilevel

    d. Hierarchical
    e. Hybrid
10. Study of Constructor & Method Overriding
11. Study of Method & Operator Overloading
12. Study of Database Connectivity
    a. Testing MySQL Database connection & Creating Cursor
    b. Create Database, Tables & Insert, Read records from it
    c. Update, Delete Records from MySQL Database using Python
    d. Parameterized Query
13. Any Five Program To implement GUI using tkinter

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester I
Course Title: Mobile Application Development

Course Code: P-MAD-133 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To learn Kotlin Programming Language
- To use Kotlin Programming Language for the development of Android Applications
- To Learn Kuddo & Nutrilicious Tools for development of Applications
- Use of DSLs for complex Application Development

Course Outcomes: After completion of this course students will be able to

- Describe the basic concepts and principles to develop the mobile application
- Develop the Android Applications using Kotlin Programming Language.
- Use Kuddo & Nutrlicious tool for the development of Android App
- Develop the complex Application Development

Practical List

1. Implement the program of mutable and Read only variables
2. Implement the program of Conditional statement in Kotlin and Construct Looping statements in Kotlin (While & For Loop)
3. Declaring and calling a simple and extension function in Kotlin
4. Implement the Nullable and Non nullable types and accessing members of nullable variables in Kotlin
5. Implement exception handling using try, catch, finally block
6. Implement the program of Inheritance and Data class in Kotlin
7. Implement the program of classes and objects in Kotlin and also implement the program of constructers in Kotlin (Primary and Secondary)
8. Create a Kuddo App – To do List in Android Studio
9. Create a Nutrilicious App – in Android Studio
10. Create Anko Layouts over hard Layouts using Kotlin in Android Studio
11. Create Weather forecasting application using Kotlin in Android Studio
12. Create a paint and calculator application using Kotlin in android studio
13. Create Quiz Application using Kotlin in Android Studio
14. Build a Planet app using Kotlin in Android Studio
15. Create a chatting Application using Kotlin in Android Studio

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur

Department of Computer Science

M. Sc. (Computer Science) First Year Semester I

Seminar I

Course Code: P-SEM-134 **Credits: 2**

Max. Marks: 25

Learning Objectives:

- To expand the subject knowledge
- To Improve the communication & Presentation skills

Course Outcomes: After completion of this course students will be able to

- Express their knowledge on the given topic in a well manner
- Present information in a compelling, well-structured, and in a logical sequence.

Steps to Complete the Seminar

1. Allotment of Seminar Topic
2. Preparation of Presentation & Delivery of Seminar (Minimum 30 Minutes)
3. Preparation of Seminar Report
4. Final Presentation & Submission of Seminar Report

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year
Question Paper Pattern for Semester I & II
With effect from June 2022

Note: All questions are compulsory.

Q. 1 Answer any three of the following questions (Each 5 Marks) 15

a. Question from Unit I
b. Question from Unit II
c. Question from Unit III
d. Question from Unit IV

Q. 2 Answer any three of the following questions (Each 10 Marks) 30

a. Question from Unit I
b. Question from Unit II
c. Question from Unit III
d. Question from Unit IV

Q.3 Write short notes on any three of the following (Each 5 Marks) 15

a.
b.
c.
d.

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Advanced DBMS

Course Code: P-ADB-225 **Credits: 4**
Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To impart the concepts of Distributed and Centralized Databases.
- To enable with the principles of Query transformation and Optimization techniques.
- To nurture with precepts of transaction management in distributed database.
- To discuss the concurrency control concepts in distributed systems.
- To familiarize the basics concepts of reliability and inconsistency problems of distributed database systems

Course Outcomes: After completion of this course students will be able to

- Solve transforming a global query into local query using optimizing techniques.
- Summarize the distributed transaction management principles.
- Explain various distributed concurrency control techniques.
- Evaluate the Non-blocking Commitment Protocols

Unit No.	Contents	No. of Lect.
Unit I Relational Databases	Structure of Relational Database, Database Schema, Keys, Schema Diagrams, Relational Operations, Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus. Overview of the design Process, Entity Relationship Model, Intraradical states, Removing Redundant attributes in entity sets, E-R Diagram, Entity Relationship Design Issues Intermediate and Advanced SQL	**15 Lectures**
Unit II Query Processing and Optimization	Measures of Query Cost, Selection Operation, Sorting, Join Operations, Other Operations Transformation of Relational Expressions, Estimating statistics of expression results, choice of evaluation plan, Materialized views, advanced topics in Query optimization.	**15 Lectures**
Unit III Transaction Management and Concurrency Control	Transaction concept, simple transaction model, Storage Structure, Transaction Atomicity and durability, Transaction isolation, Serializability, Transaction isolation and atomicity, Transaction isolation level and implementation, Transactions as SQL Statements Lock based Protocol, Deadlock handling, Time stamp-based protocol, Validation based protocol	**15 Lectures**
Unit IV System Architecture	Data base system architecture-centralized and client server architecture, Server system architecture, Parallel System and distributed systems I/O Parallelism, interquery Parallelism, Intraquery Parallelism, Intraoperation Parallelism, Interoperation Parallelism	**10 Lectures**

	Homogeneous and Heterogeneous databases, Distributed data storage, Distributed transactions, commit protocols, Concurrency control and distributed databases, Distributed query processing	

REFERENCE BOOKS:

Database system Concepts, Abrahan Silberschatz, Henry F. Korth, S. Sudarshan, Mc Graw Hills Publication

An Introduction to Database Systems, Bipin C. Desai, Galgotia Publications

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Compiler Design

Course Code: P-COD-226 **Credits: 4**
Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To get working knowledge of the major phases of compilation, like lexical analysis, parsing, semantic analysis and code generation.
- To use the formal attributed grammars for specifying the syntax and semantics of programming languages.
- To understand the structure of a compiler, and how the source and target languages influence various choices in the design
- To learn and use tools for compiler construction.

Course Outcomes: After completion of this course students will be able to

- To solve problem of parsing and compiling.
- Design and develop simple compiler.
- Use compiler tools in basic, concurrent, distributed and embedded environments.
- Generate and optimize the code.

Unit No.	Contents	No. of Lect.
UNIT I: Introduction to Compilers and Lexical Analysis	Compilers and Translators, The Structure of Compiler, Lexical Analysis, Syntax Analysis, Intermediate Code generation, Optimization, Code Generation, Bookkeeping, Error Handling, Compiler writing Tools, Programming Language Basics: Definition, The lexical and Syntactic structure of a language, data elements, data structures, operators, assignment, statements, program units, data environments, parameter transmission, storage management Lexical Analysis: Role of Lexical Analyzer, A simple approach to the design of Lexical Analyzers, Regular Expressions, Finite Automata, from regular expressions to finite automata, Minimizing the number of states of a DFA, A language specifying lexical analyzers, Implementation of a lexical analyzer, The scanner generator as Swiss army knife	**18 Lectures**
UNIT II: Syntax Analysis and Parsing Techniques	Context free grammars, Derivations and Parse Trees, Capabilities of Role of Context free grammars, Parser, shift reduce parsing, Operator Precedence Parsing, top-down parsing, Predictive parsers – Computation of FIRST & FOLLOW functions and construction of parsing table, LR parsers, the canonical collection of LR (O) items, Constructing LALR parser tables, Using Ambiguous Grammars, An Automatic parser Generator, Implementation of LR parsing tables, Constructing LALR sets of items	**12 Lectures**

UNIT III: Syntax Directed Translation and Symbol Table	Syntax-Directed Translation schemes, Implementation of syntax-directed translators, Intermediate code, Postfix notations, parser trees and syntax trees, three address codes – Quadruples and triples, indirect triples, Translation of assignment statements, Boolean expressions, Statements that alter the flow of control, Postfix translations, Translation with a top-down parser Symbol Tables: The Contents of Symbol Table, Data Structures for a Symbol Tables, Representing scope information	**15 Lectures**
UNIT IV: Code Optimization and Code Generation	The principal sources of optimization, loop optimization - Basic blocks, flow graphs, loops, code motion, induction variables, The DAG representation of basic blocks-Application of DAGs, Value Numbers and Algebraic Laws, Global Data Flow Analysis-Data Flow equations, Solving Data Flow equations. Object programs: the environment of code, generator, run-time addresses for names, Problems in code generation, A machine model, a simple code generator, Register allocation and assignments, Code generation from DAG's, Peephole optimization. Introduction to Errors, Lexical Phase Errors, Syntactic Phase Errors, Semantic Phase Errors	**15 Lectures**

REFERENCE BOOKS:

1. Principles of Compiler Design - Alfred V.Aho, Jeffey D. Ullman (Narosa Publishing House)
2. Compiler Construction – Principles & Practices D. M. Dhamdhere,
3. Compilers Principles, Techniques and Tools Alfred V. Aho Second Edition (Pearson Education)

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Web Programming

Course Code: P-WPR-227 **Credits: 4**
Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- ASP.NET helps students to create their own web applications.
- To design & develop static and dynamic websites.
- To develop web application with validation controls
- To train the students in creating dynamic web pages using ASP.NET.
- To facilitate the students, develop real time applications using database.

Course Outcomes: After completion of this course students will be able to

- Develop web pages using HTML, CSS and JavaScript.
- 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.
- Build web applications using ASP.NET and MySQL database / MS-Access.

Unit No.	Contents	No. of Lect.
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, Directive	**15 Lectures**
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 Lectures**
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 Controls of Master page in Content page, Master page with Menus.	**15 Lectures**
Unit-IV: User Controls,	Creating User Control, Required Field Validator, Compare Validator, Range Validator, Regular Expression Validator,	**15 Lectures**

Validation, State Management and Web Services	Custom Validator, Query String, State Management, Hidden Field, Cookies, Session, Creating Web Services, Web Methods, Database Oriented Asp.net, ADO.NET data access, Data Binding, Web Application with Grid View, Data List, Data Grid, Repeater.	

REFERENCE BOOKS:

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

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Internet of Things

Course Code: P-INT-228 **Credits: 4**
Total Lectures: 60 **Max. Marks: 100**

Learning Objectives:

- To introduce the concepts of Internet of Things.
- To impart the knowledge on IoT application areas.
- To introduce the IoT business process models, design technology for Connected Devices.
- To enable the students, learn the effective usage of device connectivity and web connectivity models.

Course Outcomes: After completion of this course students will be able to

- Demonstrate the need of IoT in the computing world.
- Identify the Business Process models of IoT.
- Analyze the data storage and acquisition mechanisms for real time applications.
- Design IoT based prototypes.

Unit No.	Contents	No. of Lect.
Unit-I: Introduction Internet of Things	Definition and characteristics of IoT. Sensing, Actuation, Networking basics, Sensor Network. 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 Lectures**
Unit-II: Domain Specific IoTs and IOT vs M2M	**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 **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. **IOT vs M2M: -** M2M, Difference between IoT and M2M, Difference between SDN and NFV for IoT- software defined	**15 Lectures**

	networking and network function virtualization, IoT Code generator. An emerging industrial structure for IOT, Use case example.	
Unit-III: 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, Case Study on IoT System for weather monitoring.	**15 Lectures**
Unit-IV: Developing IoT Solutions.	What is an IoT Device? , Raspberry Pi Interfaces, Programming Raspberry Pi with Python, Other IoT Devices, IoT Physical Servers on Cloud Offering, Amazon Web Services for IoT, Case studies- Home Automation , Cities, Environment, Agriculture.	**15 Lectures**

REFERENCE BOOKS:

1. Internet of Things (A hands on approach) - Vijay Madisetti and Arshdeep Bagha
2. Designing the Internet of Things - Adrian McEwen & Hakim Cassimally.
3. Rethinking the Internet of Things - A scalable approach to connecting everything - Francis Dacosta.

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Advanced DBMS

Course Code: P-LAC-229 **Credits: 2**

Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To impart the concepts of Distributed and Centralized Databases.
- To enable with the principles of Query transformation and Optimization techniques.
- To nurture with precepts of transaction management in distributed database.
- To discuss the concurrency control concepts in distributed systems.
- To familiarize the basics concepts of reliability and inconsistency problems of distributed database systems

Course Outcomes: After completion of this course students will be able to

- Solve transforming a global query into local query using optimizing techniques.
- Summarize the distributed transaction management principles.
- Explain various distributed concurrency control techniques.
- Evaluate the Non-blocking Commitment Protocols

PRACTICAL LIST

1. Introduction SQL-SQL*Plus
    a. My SQL Installation
    b. Types of SQL Commands
    c. Various Data Types
    d. Database & Tables Creation
    e. Key Constrains-Normalization
    f. Update, Delete, Alter , Rename
    g. where, In, not in, like, not like, distinct, is null, is not null
2. Working with sorting, grouping & Aggregate functions
    a. Order by Clause
    b. Group by Clause, Having Clause
    c. Single Row Functions: character, number, date etc.
    d. Multi-row Functions
3. Working with Table Join
    a. Cartesian Product
    b. ANSI Style
    c. Using Clause
    d. Theta Join
    e. Outer Join- Left, Right, Full Outer Join
    f. Self-join
    g. Set Operators – Union, Intersect, Minus

4. Working with Views in Oracle 10g
5. Programs on PL/SQL Block in ORACLE 10g
    a. Simple PL/SQL Block
    b. Looping in PL/SQL Block
    c. Exception Handling in PL/SQL
6. Programs on Trigger, Procedure, Cursor in Oracle 10g
    a. Row Triggers and Statement Triggers
    b. Before and After Triggers
    c. Instead Of Triggers
    d. Implicit cursors
    e. Explicit cursors
    f. Procedures –In, Out, In Out
7. Working with Database Transactions in MySQL/ORACLE/ PostgreSQL
    a. Set transaction
    b. Begin transaction
    c. End transaction
    d. Commit, Rollback, Savepoint
8. Program on Distributed Transactions in PostgreSQL

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Compiler Design

Course Code: P-LAC-230 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To get working knowledge of the major phases of compilation, like lexical analysis, parsing, semantic analysis and code generation.
- To use the formal attributed grammars for specifying the syntax and semantics of programming languages.
- To understand the structure of a compiler, and how the source and target languages influence various choices in the design
- To learn and use tools for compiler construction.

Course Outcomes: After completion of this course students will be able to

- To solve problem of parsing and compiling.
- Design and develop simple compiler.
- Use compiler tools in basic, concurrent, distributed and embedded environments.
- Generate and optimize the code.

List of Practical:

1. Tokenizing a file using C.
2. Implementation of Lexical Analyzer using Lex Tool.
3. Study the LEX and YACC tool and evaluate an arithmetic expression with parentheses, unary and binary operators using Flex and Yacc (CALCULATOR).
4. Using JFLAP, create a DFA from a given regular expression.
5. Create LL(1) parse table for a given CFG and hence Simulate LL(1) Parsing.
6. Using JFLAP create SLR(1) parse table for a given grammar. Simulate parsing and output the parse tree proper format.
7. Write functions to find FIRST and FOLLOW of all the variables.
8. Read a regular expression in its standard form and find out an Ɛ-NFA from it. Need to use adjacency list data structure of graph to store NFA. Thompson's construction needs to be used too.
9. Read a regular expression in standard form and check its validity by converting it to postfix form. Scan a string and check whether the string matches against the given regular expression or not.
10. Design predictive parser for the given language.
11. Implementation of shift reduce parsing algorithm.
12. Convert the BNF rules into YACC form and write code to generate abstract syntax tree.

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year Semester II
Course Title: Web Programming

Course Code: P-LAC-231 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- ASP.NET helps students to create their own web applications.
- To design & develop static and dynamic websites.
- To develop web application with validation controls
- To train the students in creating dynamic web pages using ASP.NET.
- To facilitate the students, develop real time applications using database.

Course Outcomes: After completion of this course students will be able to

- Develop web pages using HTML, CSS and JavaScript.
- 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.
- Build web applications using ASP.NET and MySQL database / MS-Access.

List of Practical:

1. Introduction and Installation of Visual Studio.
2. Program for variables declaration and operators in Asp.net.
3. Program for Decision Making, Loops and Function in Asp.net.
4. Program to demonstrate Label, TextBox, Button Control.
5. Program to demonstrate ListBox, ComboBox Control.
6. Program to demonstrate Dropdown list and Calendar Control.
7. Program to design a Masterpage.
8. Program for embedding CSS in Asp.net.
9. Program to demonstrate Exception Handling in Asp.Net.
10. Program to demonstrate Compare validator, Required Filed Validator.
11. Program to demonstrate Range Validator and Custom Validator.
12. Program on Cookies.
13. Program on Session.
14. Program to demonstrate View State and Query String.
15. Database Connectivity.

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
M. Sc. (Computer Science) First Year Semester II
Course Title: Internet of Things

Course Code: P-LAC-232 **Credits: 2**
Total Lectures: 45 **Max. Marks: 50**

Learning Objectives:

- To introduce the concepts of Internet of Things.
- To impart the knowledge on IoT application areas.
- To introduce the IoT business process models, design technology for Connected Devices.
- To enable the students, learn the effective usage of device connectivity and web connectivity models.

Course Outcomes: After completion of this course students will be able to

- Demonstrate the need of IoT in the computing world.
- Identify the Business Process models of IoT.
- Analyze the data storage and acquisition mechanisms for real time applications.
- Design IoT based prototypes.

List of Practical:

1. Install Virtual box and Raspberry Pi to perform actions of Raspberry Pi.
2. Starting Raspbian OS, familiarizing with raspberry pi components and Interface, connecting to ethernet, monitor, USB.
3. Displaying different LED patterns with Raspberry Pi.
4. Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi.
5. Control Raspberry Pi via Telegram Messenger.
6. Setting up Wireless Access Point using Raspberry Pi.
7. Fingerprint Sensor interfacing with Raspberry Pi.
8. GPS Module Interfacing with Raspberry Pi.
9. IOT based Web Controlled Home Automation using Raspberry Pi
10. Visitor Monitoring with Raspberry Pi and Pi camera
11. RFID interfacing with Raspberry Pi.
12. Building Google Assistant with Raspberry Pi.
13. Installing windows 10 IoT core on Raspberry Pi.
14. Light on LED through Python Program using Raspberry Pi
15. Get input from switches and switch on corresponding LEDs controlling on LEDs by button.

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur

Department of Computer Science

M. Sc. (Computer Science) First Year Semester II

Seminar II

Course Code: P-SEM-234 **Credits: 2**

Max. Marks: 25

Learning Objectives:

- To expand the subject knowledge
- To Improve the communication & Presentation skills

Course Outcomes: After completion of this course students will be able to

- Express their knowledge on the given topic in a well manner
- Present information in a compelling, well-structured, and in a logical sequence.

Steps to Complete the Seminar

5. Allotment of Seminar Topic
6. Preparation of Presentation & Delivery of Seminar (Minimum 30 Minutes)
7. Preparation of Seminar Report
8. Final Presentation & Submission of Seminar Report

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Computer Science
M. Sc. (Computer Science) First Year
Question Paper Pattern for Semester I & II
With effect from June 2022

Note: All questions are compulsory.

Q. 1 Answer any three of the following questions (Each 5 Marks) 15

a. Question from Unit I
b. Question from Unit II
c. Question from Unit III
d. Question from Unit IV

Q. 2 Answer any three of the following questions (Each 10 Marks) 30

a. Question from Unit I
b. Question from Unit II
c. Question from Unit III
d. Question from Unit IV

Q.3 Write short notes on any three of the following (Each 5 Marks) 15

a.
b.
c.
d.