Rajarshi Shahu Mahavidyalaya, Latur
(Autonomous)

Department of Computer Science and Information Technology

Syllabus for Post Graduation Course (M.Sc.C.S.)

(With Effect from Academic Year – 2017-2018 Under CBCS)

Name of Programme: Master of Computer Science (First Year Semester I + II)

Semester : I

Code No	Course Name	Maximum Marks		Total Marks	Credits
		Theory / Practical	Internal		
P-DAA-126	Design and Analysis of Algorithm	60	40	100	4
P-OUP-128	OOP's Using Python	60	40	100	4
P-MOC-129	Mobile Computing	60	40	100	4
P-DAM-130	Data Mining	60	40	100	4
P-LAC-131	Lab Course – I (DAA)	100	--	100	4
P-LAC-132	Lab Course – II (Python)	100	--	100	4

Semester : II

Code No	Course Name	Maximum Marks		Total Marks	Credits
		Theory / Practical	Internal		
P-NUM-225	Numerical Methods	60	40	100	4
P-COD-226	Compiler Design	60	40	100	4
P-WPU-227	Web Programming using ASP.Net	60	40	100	4
P-INT-228-I	Internet of Things	60	40	100	4
P-BIO-228-B	Bioinformatics	60	40	100	4
P-LAC-229	Lab Course – III (NM)	100	--	100	4
P-LAC-230	Lab Course – IV (Asp.Net)	100	--	100	4

P-DAA-126

DESIGN AND ANALYSIS OF ALGORITHM

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

This is a first course in data structures and algorithm design. Students will: learn good principles of algorithm design;learn how to analysis algorithms and estimate their worst-case and average-case behavior (in easy cases);become familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles; learn how to apply their theoretical knowledge in practice (via the practical component of the course).

Course Outcome:

Basic strategies of algorithm design: top-down design, divide and conquer, average and worst-case criteria. Simple recurrence relations for asymptotic costs. Choice of appropriate data structures: arrays, lists, stacks, queues, trees, heaps, priority queues, graphs, hash tables. Applications to sorting and searching, matrix algorithms, shortest-path and spanning tree problems. Introduction to discrete optimisation algorithms: dynamic programming, greedy algorithms. Graph algorithms: depth first and breadth first search.

Syllabus

UNIT I: Introduction

A simple example of design using insertion sort, pseudo code for insertion sort and analysis of time complexity. Performance Analysis – Space complexity and Time complexity (posteriori testing, and apriori approach), Asymptotic Notations (O, Ω, Θ). Polynomial vs. Exponential Algorithms. Average, Best and Worst case complexity.

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 Algorithms, Optimal Merge patterns and Single-Source Shortest Paths.

UNIT III: Dynamic Programming and Back tracking and Branch and 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

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

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. GAV PAI, "Data structures and Algorithms", Tata McGraw Hill
5. Richard F. Gilberg, Behrouz A, Forouzan "Data structures A Pseudocode Approach with C"

P-CSA-127

DESIGN AND ANALYSIS OF ALGORITHM

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:
By learning this subject student got overall knowledge of processor design.

Course Outcome:

Students are able to do the programming of 8086 processor with deeper knowledge of processor working and architecture.

Syllabus

UNIT I: Design Methodology& Processors Design

Introduction to system modeling design levels of combinational And
Sequential
Circuit design Register level design, Register transfer languages
Design methods at processor level components
Design techniques
Queuing
Models
Simulation
Processor organization
Information representation
RISC, CICS, and Vector processor concepts.

UNIT II:Arithmetic Operation&Control Units
Fixed point arithmetic
Algorithms for addition subtraction, multiplication & division
Floating point arithmetic.
Hardwired control units organization and operation
Micro-programmed control units
Micro program decoder sequencer
Interrupt and branch instruction processing
Instruction sequencing interpretation.

UNIT III: Memory Organization and Introduction to Microprocessor

Virtual memory, Memory hierarchies, main memory allocation
Segmentation, paging paged segmentation.
High speed memories interleaved memories, associative memories.
Introduction to Microprocessor
History of Microprocessor
Study of Intel Processor Family
Introduction to I86, 386, Dual Core, Quadcore
Introduction to i3, i5 processor

UNIT IV: **8086 Architecture and Instruction Set**

Introduction to 8086 Processor
8086 CPU Architecture.
EU & BIU activities,
Segmentation and address transition
8086 pin description
Addressing modes
Data Transfer, arithmetic logical string, i/o instruction
Control group of instruction
Assembly Language Programming of 8086.

Reference Books:

1. Computer Architecture & Organization -J. P. Hayes. (MGH)

2. Computer Architecture & Organization -J. P. Hayes. (MGH)

3.8086/8088 Microprocessor Family –Liu Gibson (MGH)

4. Microprocessor Architecture Programming andApplication- Ramesh

Gaonkar, Willey Estern.

P-OUP-128

OOP's USING PYTHON

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

- This is an introductory course designed for any student interested in using computation to enhance their problem solving abilities.
- Students will use their problem solving abilities to implement problems in python.
- No prior programming experience is necessary
- Develop a basic understanding of programming and python programming language

Course Outcome:

After completion of this course, students should be able to:

- Understand and write python programs.
- Implement object oriented concepts in real sense
- Can be easily work on any platform
- Utilize the distinct features provided by python

Syllabus

UNIT- I : Beginning Python

Introduction, History, Important Features, Overview of python and installation, Lexical Matters: lines, Comments, Names and Tokens, Doc Strings.

Unit II : Getting Started, Simple program, Identifiers, Reserved words, Multi-Line Statements, Operators, Variables, Assignment, Numbers(Int, long, float and complex), Strings, Decision and Looping statements: Introduction to decision statement, If – else statement, Introduction to looping statement, While loop, For loop, Nesting loop, Break, Continue and pass statement

Unit – III : Sequence:

Strings , Strings and operators, String built-in methods, Lists, List type built-in method, Tuples

Special features of tuples, What are exceptions, Exceptions in python, Detecting and handling exceptions, Raising exception, Assertions, Standard exceptions, Creating exceptions

Unit – IV: Functions, class and OOPs

What are functions, Calling functions, Creating functions, Passing functions, Formal arguments positional arguments, default arguments, Variable length argument, Recursion, Introduction: OOP, Classes, Class attributes, Instances, Instance attribute, Building and method invocation, Sub classing and derivation, Inheritance, Built-in functions for classes, Instances and other objects, privacy,

Reference books:

1. Core Python Programming, Second Edition, by Welesy J. Chun.
2. Programming Python, OREILLY Publicatgion, By Mark Lutz.
3. A Python Book: Beginning Python, Advanced Python, and Python Exercises, by Dave Kuhlman

P-MOC-129

MOBILE COMPUTING

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

This course introduces the basic concepts and principles in mobile computing. This includes the major techniques involved, and networks & systems issues for the design and implementation of mobile computing systems and applications. This course also provides an opportunity for students to understand the key components and technologies involved and to gain hands-on experiences in building mobile applications.

Course Outcome:

Describe the basic concepts and principles in mobile computing.

Understand the concept of Wireless LANs, PAN, Mobile Networks, and Sensor Networks.

Explain the structure and components for Mobile IP and Mobility Management

Understand positioning techniques and location-based services and applications

Describe the important issues and concerns on security and privacy.

Syllabus

Unit I: Introduction to GSM **20Hrs**

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

Unit II: Medium Access Control (MAC) **10Hrs**

Motivation for specialized MAC

SDMA

FDMA

TDMA

CDMA

Unit III: Telecommunication System and GSM **20Hrs**

Mobile services

System architecture

Radio interface

DECT

System architecture

Protocol architecture

TETRA

UMTS and IMT-2000

Unit IV: Wireless LAN **10Hrs**

Infra red Vs radio transmission

Infrastructure and along Network

IEEE 802.11, HIPERLAN, Bluetooth

Reference Books:

1. Mobile Communications Second Edition – By Jochen Schiller (Pearson Education)

2. Mobile Cellular Telecommunications Second Edition-ByWilliamC.Y.Lee (Mc-Graw-Hill)

P-DAM-130

DATA MINING

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

- 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..

Course Outcome:

- Display a comprehensive understanding of different data mining tasks and the algorithms most appropriate for addressing them.
- 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.

Syllabus

Unit I: 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.

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, Distance Based Algorithms, Decision Tree-Based Algorithm: CART, Neural Network-Based Algorithm: NN Supervised Learning.

Unit III: Clustering and Association Rules

-Introduction to Clustering, outliers, K-Means clustering, Nearest Neighbor Algorithm, BRICH algorithm.

-Introduction to Association Rules, Large Itemsets, Basic Algorithms: Apriori Algorithm, Data Parallelism, Comparing Approaches.

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.

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 Course -1
P-LAC-131

DESIGN AND ANALYSIS OF ALGORITHM

Total Credits: 4

Learning Objective:

This is a first course in data structures and algorithm design. Students will: learn good principles of algorithm design; learn how to analysis algorithms and estimate their worst-case and average-case behavior (in easy cases);become familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles; learn how to apply their theoretical knowledge in practice (via the practical component of the course).

Course Outcome:

Basic strategies of algorithm design: top-down design, divide and conquer, average and worst-case criteria. Simple recurrence relations for asymptotic costs. Choice of appropriate data structures: arrays, lists, stacks, queues, trees, heaps, priority queues, graphs, hash tables. Applications to sorting and searching, matrix algorithms, shortest-path and spanning tree problems. Introduction to discrete optimization algorithms: dynamic programming, greedy algorithms. Graph algorithms: depth first and breadth first search.

Proposed Practical List:

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

M. Sc. CS Lab Course -2
P-LAC-132

OOP's USING PYTHON

Total Credits: 4

Learning Objective:

- This is an introductory course designed for any student interested in using computation to enhance their problem solving abilities.
- Students will use their problem solving abilities to implement problems in python.
- No prior programming experience is necessary
- Develop a basic understanding of programming and python programming language

Course Outcome:

After completion of this course, students should be able to:

- Understand and write python programs.
- Implement object oriented concepts in real sense
- Can be easily work on any platform
- Utilize the distinct features provided by python

Proposed Practical List:

1. Calculate the average numbers in a given list.
2. Reverse a given number
3. Take in the marks of five subjects and display the grade
4. Accept three digits and print all possible combinations from the digits
5. Find the sum of digits in a number
6. Put even and odd elements in a list into two different lists
7. Find the second largest number in a list using bubble sort
8. Replace all occurrences of 'a' with '$' in a string
9. Count number of lowercase characters in a string
10. Calculate the number of digits and letters in a string
11. Add a key-value pair to the dictionary
12. Determine how many times a given letter occurs in a string recursively
13. Find the area of rectangle using classes

P-NUM-225
NUMERICAL METHOD

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

- 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.

Course Outcome:

- 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.

Syllabus

Unit I Computer Arithmetic & Solution of algebraic equations

Computer Arithmetic

1. Floating Point representation of Numbers
2. Arithmetic operation with Normalized floating point

Solution of algebraic equations

Bisection method

Method of false position

Newton-Raphson Method

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

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

Unit IV Curve Fitting

Least square Curve fitting

Fitting a straight line

Non linear curve fitting polynomial of n^{th} degree

Reference Books

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

P-COD-226
COMPILER DESIGN

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

- To introduce the major concept areas of language translation and compiler design
- To develop an awareness of the function and complexity of modern compilers.
- To introduce various phases of compiler design.

Course Outcomes:

After completion of this course, students should be able to:

- Students will have a concrete view on the theoretical and practical aspects of compiler design
- Students will be able to apply ideas and techniques discussed to various software design.

Syllabus

UNIT - I Programming Languages and Compilers

Introduction to Compilers
Compilers and translators
The structure of compiler
Compiler writing tools
High level programming languages
Definitions of programming languages
A lexical and syntactic structure of a language
Data structures
Operators
Statements

UNIT – II Lexical Analysis

Lexical analysis
Role of a Lexical analyzer
A simple approach to the design of lexical analyzer
Regular expressions
Finite automata
Minimizing number of states of a DFA
Implementation of a lexical analyzer

UNIT – II Basic Parsing Techniques and Syntax Directed Translation
Context free grammars
Introduction to parsers
Shift reduce parsing
Top-down parsing
Operator Precedence parsing
Predictive parsers
Introduction Syntax Directed Translation
Syntax directed Schemes
Implementation of Syntax directed translators
Intermediate code
Postfix notation and evaluation of postfix expressions
Parse trees and syntax trees

UNIT – IV Symbol Tables, Errors and Code Optimization
The contents of a symbol table
Data structures for a symbol table
Errors
Lexical-phase errors
Syntactic phase errors
Semantic errors
Introduction Code Optimization
Sources of optimization
Loop Optimization

Reference books:

1. Principles of Compiler Design- By Alfred V. Aho, Jeffrey D. Ullman. Narosa Publishing House ISBN-81-85015-61-9

2. Compilers ,Principles, Techniques and Tools - A.V. Aho, Ravi Sethi and J.D. Ullman. ISBN-817-808-046-x

3. Introduction to system software By D. M. Dhamdhere

P-WPU-227
Web Programming using Asp.net

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

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.

Course Outcomes:

- 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.

Syllabus

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 , OS role in web development.
- Clientside and Serverside 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

Unit-II: Asp.net Applications, CSS and Themes

- Creating Asp.net Web Application
- Auto Postback 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

Unit-III: Redirecting User to Another Page & Master Pages.

- Redirecting Options
- Response.Redirect
- Server.Transfer
- Cross Page Postback
- Passing Values between pages
- Introduction to Master Page
- ContentPlaceHolder and Content tags
- Accessing Controls of Masterpage in Content page
- Masterpage 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, ViewState, HiddenField
- Cookies
- Session
- Creating Web Services
- Web Methods
- Database Oriented Asp.net Web Application with GridView

Reference Books:-

1] Asp.net 4.0 Black Book

2] Mastering Asp.net , BPB Publication, Russel.

3] Asp.net the Complete Reference: MatThew Macdonald

P-INT-228-I

Internet of Things

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

- 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.

Course Outcomes:

- Able to understand the application areas of IOT
- Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
- Able to understand building blocks of Internet of Things and characteristics.

Syllabus

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

IoTenabling Technologies-Wireless sensor networks, cloud computing, big data analytics, communication protocols, embedded systems

IoTLevels and deployment templates-IoT Level1 to IoT Level6

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

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

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

Reference Books:-

1] Internet Of Things (A hands on approach) By: Vijay Madisetti and Arshdeep 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

P-BIO-228-B

BIOINFORMATICS

Total Teaching Hours: 60 **Total Credits: 4**

Learning Objective:

Bioinformatics is the science of storing, extracting, organizing, analyzing, interpreting and using information. The approaches to the discipline of bioinformatics incorporate expertise from the biological sciences, computer science and mathematics. The major in bioinformatics is designed for students interested in molecular biology and genetics, information technologies and computer science. Bioinformaticists are involved in the analysis of the human genome, identification of targets for drug discovery, development of new algorithms and analysis methods, the study of structural and functional relationships, and molecular evolution.

Course Outcomes:

- Knowledge and awareness of the basic principles and concepts of biology, computer science and mathematics
- Existing software effectively to extract information from large databases and to use this information in computer modeling
- Problem-solving skills, including the ability to develop new algorithms and analysis methods
- An understanding of the intersection of life and information sciences, the core of shared concepts, language and skills the ability to speak the language of structure-function relationships, information theory, gene expression, and database queries

---

Syllabus

Unit-I : Bioinformatics: An introduction, Information Search and Data Retrieval

Introduction to bioinformatics, Historical overview and definition, Applications, Major databases, Introduction to information search, Tools for web search, Data retrieval tools, Data mining of biological databases.

Unit-II: GENOME Analysis and gene mapping, alignments of pairs of sequences

GENOME Analysis, GENOME Mapping, The sequence Assembly problems, Physical maps, Application of genetics maps, Biological motivations of alignment problems, Methods of sequence alignments, Using scoring matrices.

Unit-III: Tools for similarity search, sequence alignment and introduction to drug discovery

Introduction search, Working with FASTA, Working with BLASTA, FASTA and BLASTA algorithm comparison, Introduction to drug discovery, Areas influencing drug discovery, Phamacogenetics and Pharmacogenomics, Applications, Important parameters in drug discovery.

UNIT-IV: Drug discovery: Technology, strategies and computer- Aided Drug Design.

Drug Discovery Technologies, Target discovery strategies, Strategy to identify possible drug targets, Target validation, Introduction to drug design, Drug design approach, Computer aided drug designing methods.

Reference Books:

1. Bioinformatics methods and Applications By S.C. Rastogi.
2. An Introduction to bioinformatics By V. Kothekar and T. Nandi, Duck Worth Press.

M. Sc. CS Lab Course -III
P-LAC-229

NUMERICAL METHODS

Total Credits: 4

Learning Objective:

- 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.

Course Outcome:

- 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.

Proposed Practical List:

1. MATLAB Environment.
2. Write a program in MATLAB for simple calculations
3. Write a program in MATLAB for dot operators
4. Write a program in MATLAB for vector indexing
5. Write a program in MATLAB for matrix indexing
6. Write a program in MATLAB for M file
7. Write a program in MATLAB for function
8. Write a program in MATLAB for average calculation
9. Write a program in MATLAB for matrix addition
10. Write a program in MATLAB for matrix multiplication
11. Write a program in MATLAB for matrix transpose
12. Write a program in MATLAB for differentiation
13. Write a program in MATLAB for integration
14. Write a program in MATLAB for error calculation

15. Write a program in MATLAB for successive bisection method.
16. Write a program in MATLAB for Newton Raphson method.
17. Write a program in MATLAB for false position method.

M. Sc. CS Lab Course -IV
P-LAC-230

Web Programming using Asp.net

Total Credits: 4

Learning Objective:

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.

Course Outcomes:

- 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.

Proposed Practical List:

1] Introduction to Visual Studio IDE

2] Creating Empty Web Site

3] Creating Asp.net Web Application

4] Asp.net application to demonstrate asp.net page life cycle

5] Asp.net application to demonstrate IsPostBack property

6] Asp.net application to demonstrate Response.Redirect

7] Asp.net application to demonstrate Server.Transfer

8] Asp.net application to works with Button and TextBox control

9] Asp.net application with try, catch and finally blocks

10] Create Webpage with CSS style

11] Creating Theme in asp.net

12] Asp.net Application with master and child pages

13] Asp.net application with Validation Controls

14] Build asp.net application to maintain state using session

15] Create Database oriented Asp.net web application