Savitribai Phule Pune University

(Formerly University of Pune)

Two Year Degree Program in

‘Industrial Mathematics with Computer Application’

(Faculty of Science & Technology)

Revised Syllabi for

M.Sc. (Industrial Mathematics with Computer Application)

Part-I

(For Colleges Affiliated to Savitribai Phule Pune University)

Choice Based Credit System Syllabus

To be implemented from Academic Year 2019-2020

Title of the Course: M. Sc. (Industrial Mathematics with Computer Application)

Preamble :

SavitribaiPhule Pune University has decided to change the syllabi of various faculties from June,2019.M.Sc. (Industrial Mathematics with Computer Applications) course syllabus is revised tocater to the needs of credit based-semester and grading system. The changing scenario ofhigher education in India and abroad is taken into consideration to make this syllabusmore oriented towards the applications of Mathematics and Computer Science inResearch and Industry. The syllabus encompasses the subjects related to IndustrialMathematics, Core Computer Subjects as well as the Emerging Technologies inComputer Science. Theory Courses will create the foundation for the development oflogical thinking and the Practical Courses gives hands on experience towards theIndustrial Requirements.

Taking into consideration the rapid changes in science and technology and new approaches in different areas of Mathematics and related subjects, Board of Studies inMathematics with consent of teachers of Mathematics and Computer Science from different colleges affiliated to University of Pune has prepared the syllabus of **M.Sc. (Industrial Mathematics with Computer Application).**

The course will follow the credit system of the Savitribai Phule Pune University.

To develop the syllabus the U.G.C. Model curriculums followed.

Aims:

i) Give the students sufficient knowledge of fundamental principles, methods and a clear perception of the innumerous power of mathematical ideas and tools and knowledge of how to use them by modeling, solving and interpreting.
ii) Reflecting the broad nature of the subject and developing mathematical tools for continuing further study in various fields of science.
iii) Enhancing students' overall development and to equip them with mathematical modeling abilities, problem solving skill, creative talent and power of communication necessary for various kinds of employment
iv) Enabling students to develop a positive attitude towards mathematics as an interesting and valuable subject of study.

Objectives:

i) A student should be able to recall basic facts about mathematics and should be able to display knowledge of conventions such annotations, terminology and recognize basic geometrical figures and graphical displays, state important facts resulting from their studies.
ii) A student should get a relational understanding of mathematical concepts and concerned structures, and should be able to follow the patterns involved with

mathematical reasoning.

iii) A student should get adequate exposure to global and local concerns so as to explore many aspects of Mathematical Sciences.

iv) Students should be able to apply their skills and knowledge, that is, translate information presented verbally into mathematical form, select and use appropriate mathematical formulae or techniques in order to process the information and draw the relevant conclusion.

v) A student should be made aware of history of mathematics and hence of its past, present and future role as part of our culture.

vi) A student should be able to write necessary algorithms and programs in different languages as per the need of the industry

<u>Course Outcome:</u>

1. an ability to apply knowledge of mathematics and computer science in practice
2. an ability to identify, critically analyze, formulate and solve problems with comprehensive knowledge in the area of specialization
3. an ability to contribute by research and innovation to solve real life problems
4. an ability to devise and conduct experiments, interpret data and provide well informed conclusions
5. an ability to function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with positive attitude
6. an ability to communicate effectively
7. an ability to appreciate the importance of goal setting and to recognize the need for life-long reflective learning

Structure of the course:

Semester – I

T/P: Theory/Practical

T/P	Code	Course Title	% of Assesment		Total Marks	Hours/Week	No of Credits
			IA	UE			
T	MIM 101	Real Analysis	50	50	100	4	4
T	MIM 102	Linear Algebra	50	50	100	4	4
T	MIM 103	C Programming	50	50	100	4	4
T	MIM 104	DBMS	50	50	100	4	4
P	MIM 105	Lab Work	50	50	100	4	4

Semester – II

T/P: Theory/Practical

T/P	Code	Course Title	% of Assesment		Total Marks	Hours/Week	No of Credits
			IA	UE			
T	MIM 201	Complex Analysis	50	50	100	4	4
T	MIM 202	Discrete Mathematical Structures	50	50	100	4	4
T	MIM 203	Data Structures	50	50	100	4	4
T	MIM 204	Software Engineering	50	50	100	4	4
T	MIM 205	Java	50	50	100	4	4
P	MIM 206	Lab Work	50	50	100	4	4

Equivalence of previous syllabus with new syllabus:

Old course	**Equivalent new course**
Semester –I	
MIM-101 Real Analysis	MIM-101 Real Analysis
MIM-102 Linear Algebra and computational geometry	MIM-102 Linear Algebra
MIM-103 Discrete Mathematical Structure	MIM-103 C Programming
MIM-104 C Programming	MIM-104 DBMS
MIM-105 Elements of Information Technology	MIM-105 Lab work
6. MIM-106 Lab work	
Semester -II	
MIM-201 Complex Analysis	MIM-201 Complex Analysis
MIM-202 Algebra	MIM-202Discrete Mathematical Structures
MIM-203 Numerical Analysis	MIM-203 Data Structures
MIM-204 Object oriented programming with C++	MIM-204 Software Engineering
MIM-205 Data Structure Using C	MIM-205Java
MIM-206 Lab work	MIM-206 Lab work

Proposed Structure of M. Sc.(IMCA)Course

Semester – I

T/P: Theory/Practical

T/P	Code	Course Title	Hours/Week	No of Credits
T	MIM 101	Real Analysis	4	4
T	MIM 102	Linear Algebra	4	4
T	MIM 103	C Programming	4	4
T	MIM 104	Database Management Systems	4	4
P	MIM 105	Lab Work	4	4

Semester – II

T/P: Theory/Practical

T/P	Code	Course Title	Hours/Week	No of Credits
T	MIM 201	Complex Analysis	4	4
T	MIM 202	Discrete Mathematical Structures	4	4
T	MIM 203	Data Structures	4	4
T	MIM 204	Software Engineering	4	4
T	MIM 205	Java	4	4
P	MIM 206	Lab Work	4	4

Semester – III

T/P/E: Theory/Practical/Elective

T/P	Code	Course Title	Hours/Week	No of Credits
T	MIM 301	Operational Research	4	4
T	MIM 302	Algebra	4	4
T	MIM 303	Advanced Java	4	4
T	MIM 304	Operating system	4	4
P	MIM 305	Lab Work	4	4
E	MIM 306	Computer Networks	2	2
E	MIM 307	R Programming for basic Data Analysis	2	2

Semester – IV

T/P: Theory/Practical

T/P	Code	Course Title	Hours/Week	No of Credits
T	MIM 401	Topology	4	4
T	MIM 402	Statistical Methods	4	4
T	MIM 403	Design and Analysis of Algorithms	4	4
T	MIM 405	Internet Technologies	4	4
T	MIM 405	Mobile Technologies	4	4
P	MIM 406	Lab Work	4	4

Semester – V

T/P/E: Theory/Practical/Elective

T/P	Code	Course Title	Hours/Week	No of Credits
T	MIM 501	Numerical Analysis	4	4
T	MIM 502	Coding Theory	4	4
T	MIM 503	Data AnalysisWith Python	4	4
T	MIM 504	Digital Image Processing	4	4
P	MIM 505	Lab Work	4	4
E	MIM 506	Cryptography and Network Security	2	2
E	MIM 507	Internet of Things	2	2

Semester –VI

Code	Course Title	No of Credits
MIM 601	Industrial Project	12

Details of Syllabus:

Semester - I

MIM 101: Real Analysis

Unit 1. Metric Spaces and its Topology: [10 lectures]

1.1 Metric Spaces Definition and Examples, k-cells, convex sets, open closed ball, properties

1.2 Definition: Neighborhood, limit point, isolated points, closed sets, interior points, open sets, perfect sets bounded sets, dense sets, examples and properties

1.3 Definition: Open cover, compact sets, examples and properties. Theorem of Weierstrass

1.4 Connected sets, definition of separated sets, connected sets and properties

Unit 2. Numerical Sequences and series [10 lectures]

2.1 Convergent Sequences, Definition and Examples Properties

2.2 Subsequences: Definition and properties

2.3 Cauchy Sequences: Definition, Examples and properties, Definition of complete metric space, examples, Definition of Monotonic Sequences and its properties

2.4 Upper and lower limits, Definition, examples and properties

2.5 Convergence of some special sequences

2.6 Series: Definition, examples and properties, series of non-negative terms, Cauchy’s condensation test and examples

2.7 The Number e

2.8 Root and ratio tests, examples

2.9 Power series, Definition, radius of Convergence, examples and properties

2.10 Summation by parts, absolute convergence

Unit 3. Continuity: [14 lectures]

3.1 Limits of functions: Definition, examples and properties

3.2 Continuous functions, Definition, examples and properties,

3.3 Continuity and Compactness

3.3.1 Bounded Set: Definition

3.3.2 Continuous image of a compact set is compact and related properties

3.3.3 Definition of Uniform Continuity and related properties

3.4 Continuity and Connectedness: continuous image of connected set is connected and related properties

3.5 Discontinuities, Definition, examples

3.6 Monotonic functions, Definition examples and properties

Unit 4. Differentiation: [06 lectures]

4.1 Derivative of a real function, Definition examples and properties

4.2 Mean Value Theorem

4.3 Continuity of derivatives,

4.4 Taylor's theorem

4.5 Differentiation of a vector valued function

Unit 5. Riemann Stieljes Integral: [10 lectures]

5.1 Definition and existence of the integral, related properties

5.2 Properties of the integral

5.3 Integration and differentiation

5.4 Integration of vector valued functions

Unit 6. Sequences and series of function: [10 lectures]

6.1 Discussion of main problem- with examples

6.2 Uniform convergence: Definition and properties

6.3 Uniform convergence: and continuity

6.4 Uniform convergence: and integration

6.5 Uniform convergence: and differentiation

Text Book:

Walter Rudin: Principles of Real Analysis, (3rd Edition, Tata McGraw Hill Publication)

Art. 2.15 to 2.42, 2.45 to 2.47, Art. 3.1 to 3.46, Art. 4.1 to 4.18 4.19 (Statement only), 4.22 to 4.28, 4.29 (Statement only), 5.1 to 5.12, 5.15 to 5.19, 6.1 to 6.15, 6.20, to 6.25, Art 7.1 to 7.17.

Reference Books:

1. C. C. Pugh, Real Mathematical Analysis.
2. T. M. Apostol, Mathematical Analysis
3. G. F. Simmons, Topology and Modern Analysis

MIM 102: Linear Algebra

Overview: To provide a basic thorough understanding of a core course in Linear Algebra techniques and theory; and to provide a sampler (aperitif) of applications to Computer Science.

Unit 1. General Vector Spaces: [16 lectures]

1.1 Real Vector Spaces

1.2 Subspaces

1.3 Linear Independence

1.4Basis and Dimension

1.5 Row Space, Column Space and Null Space

1.6 Rank and Nullity

Unit 2. Inner Product Spaces: [8 lectures]

2.1 Inner Products

2.2 Angle and Orthogonality in Inner Product Spaces

2.3 Orthonormal Bases: Gram-Schmidt Process

Unit 3. Eigenvalues, Eigenvectors: **[10 lectures]**

3.1 Eigenvalues and Eigenvectors

3.2 Diagonalization

3.3 Orthogonal Diagonalization

Unit 4. Linear Transformations: **[8 lectures]**

4.1 General Linear Transformations

4.2 Kernel and Range

4.3 Inverse Linear Transformations

4.4 Matrices of General Linear Transformations

Unit 5. Elementary Canonical Forms: **[10 lectures]**

5.1 Introduction

5.2 Characteristic Values

5.3 Annihilating Polynomials

5.4 Invariant Subspaces

5.5 Simultaneous Triangulation; Simultaneous Diagonalization

Unit 6. Applications of Linear Algebra: **[8 lectures]**

6.1 Geometry of Linear Operators on R2

6.2 Approximation Problems; Fourier Series

6.3 LU-Decomposition

6.4 Cryptography

Text Books:

1. **Elementary Linear Algebra (Applications Version); Howard Anton and Chris Rorres, 9th Edition, Wiley Student Edition**

   Chapter 5: Sec.5.1 to Sec.5.6

   Chapter 6: Sec. 6.1 to Sec. 6.3

   Chapter 7: Sec. 7.1 to Sec. 7.3

   Chapter 8: Sec. 8.1 to Sec. 8.4

   Chapter 9: Sec. 9.2; Sec. 9.4; Sec. 9.9

   Chapter 11: Sec. 11.16

2. **Linear Algebra; Kenneth Hoffman, Ray Kunze, 2nd Edition, PHI**

   Chapter 6: Sec. 6.1 to 6.5

Reference Books:

1. Linear Algebra and its Applications; 5th Edition, David C. Lay, Stephen R. Lay, Judi J. Mc Donald, Pearson Publications
2. Introduction to Linear Algebra; Krishnamurthy
3. Schaum's Outline of Linear Algebra; 3rd Edition

MIM 103: C Programming

Unit 1. Introduction **[02 lectures]**

Unit 2. Types, Operators, and Expressions **[08 lectures]**

2.1 Variable Names.
2.2 Data Types and Sizes.
2.3 Constants.
2.4 Declarations
2.5 Arithmetic Operators.
2.6 Relational and Logical Operators.
2.7 Type Conversions.
2.8 Increment and Decrement Operators.
2.9 Bitwise Operators.
2.10 Assignment Operators and Expressions.
2.11 Conditional Expressions.
2.12 Precedence and Order of Evaluation.

Unit 3. Control Flow **[08 lectures]**

3.1 Statements and Blocks.
3.2 If-Else.
3.3 Else-If.
3.4 Switch.
3.5 Loops- While and For.
3.6 Loops- Do-while.
3.7 Break and Continue.
3.8 Goto and Labels.

Unit 4. Functions and Program Structure **[10 lectures]**

4.1 Basics of Functions.
4.2 Functions Returning Non-integers.
4.3 External Variables.
4.4 Scope Rules.
4.5 Header Files.
4.6 Static Variables.
4.7 Register Variables.
4.8 Block Structure.
4.9 Initialization.
4.10 Recursion.
4.11 The C Preprocessor.

Unit 5. Pointers and Arrays **[12 lectures]**

5.1 Pointers and Addresses.
5.2 Pointers and Function Arguments.
5.3 Pointers and Arrays.
5.4 Address Arithmetic.
5.5 Character Pointers and Functions.
5.6 Pointer Arrays; Pointers to Pointers.
5.7 Multi-dimensional Arrays.
5.8 Initialization of Pointer Arrays.

5.9 Pointers vs. Multi-dimensional Arrays.
5.10 Command-line Arguments.
5.11 Pointers to Functions.
5.12 Complicated Declarations.

Unit 6. Structures **[10 lectures]**

6.1 Basics of Structures.
6.2 Structures and Functions.
6.3 Arrays of Structures.
6.4 Pointers to Structures.
6.5 Self-referential Structures.
6.6 Typedef.
6.7 Unions.
6.8 Bit-fields.

Unit 7. Input and Output **[10 lectures]**

7.1 Standard Input and Output.
7.2 Formated output- printf.
7.3 Variable-length Argument Lists
7.4 Formatted Input-Scanf
7.5 File Access
7.6 Error Handling-Stderr and Exit
7.7 Line Input and Output
7.8 Miscellaneous Functions

References:

1. Kernighan and Ritchie : The C Programming language.
2. Forouzan and Gilberg : Structured Programming approach using C, Thomson learning publications.
3. Herbert Schildt : Complete C Reference.

MIM 104: DBMS

Course Objectives:

1. To understand the fundamental concepts of database management. These concepts include aspects of database design, database languages, and database-system implementation
2. To provide a strong formal foundation in database concepts, technology and practice
3. To give systematic database design approaches covering modeling, conceptual design, logical design and an overview of physical design
4. Be familiar with the basic issues of transaction processing and concurrency control

Course Outcomes:

1. Design E-R Model for given requirements and convert the same into database tables.
2. Use Relational database techniques such as PgSQL& PL/SQL.

Unit 1: Introduction to DBMS **[6 Lectures]**

1.1 Overview
1.2 File system Vs DBMS
1.3 Describing & storing data (Data models (relational,hierarchical, network))
1.4 Levels of abstraction
1.5 Data independence
1.6 Queries in DBMS (SQL : DDL,DML,DCL,TCL)
1.7 Structure of DBMS, People who deal inDBMS
1.8 Advantages of DBMS

Unit 2: Conceptual Design (E-R model) **[12 Lectures]**

2.1: Overview of DB design
2.2: ER data model (entities,attributes, entity sets, relations, relationship sets)
2.3: Additional constraints (key constraints, participation constraints, weak entities, aggregation / generalization.
2.4:Conceptual design using ER (entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER)
2.5: Conceptual design for small to large enterprises,Case studies

Unit 3: Relational data model **[6 Lectures]**

3.1: Relations (concepts, definition)
3.2: Conversion of ER toRelational model , integrity constraints (key, referential integrity, general constraints)

Unit 4: Relational algebra **[8 Lectures]**

4.1: Preliminaries
4.2: Relational algebra (selection, projection,set operations, renaming, joins, division)
4.3: Advance RA operators (Aggregate operators)

Unit 5: SQL **[15 Lectures]**

5.1: DDL (create, drop, alter)
5.2:Forms of a basic SQL query (egs, expressions, strings in SQL)
5.3:Union / intersection / except, nested queries(introduction
5.4:Correlated queries
5.5:Set comparison operators)
5.6: Aggregate operators (group by, having), aggrerate functions, Null values (comparison using NULL, logical connections (AND,OR,NOT) impact on SQL commands, outer joins,disallowing NULL)
5.7: Examples on SQL (case studies)

Unit 6: Functional dependency **[8 Lectures]**

6.1: Introduction to schema refinement
6.2:Problems caused by redundancy
6.3: Use of decomposition
6.4:Problems related to decomposition
6.5:Functional dependencies
6.6: F+, attribute closure
6.7: Algorithm for identifying super keys
6.8: Normalization forms (1NF, 2NF, 3NF, BCNF),

Unit 7: PL/PgSQL **[10 Lectures]**

7.1: A PL/SQL block
7.2:Concept of a Stored Function(Creation/Updation/Deletion)
7.3: Cursors
7.4: Exception Handling
7.5: Database Triggers, Views

References:-

1. Fundamentals of Database Systems (4th Ed) By: Elmasri and Navathe
2. Database System Concepts (4th Ed) By: Korth, Sudarshan, Silberschatz
3. Practical PostgreSQL ,O’reilly Series

MIM 105: Lab Work

1. Write simple C programs area of triangle, circle, simple and compound interest, Celsius to Fahrenheit.
2. (a) Write a program to sort 5 numbers using 7 comparisons.
(b) Write a program count number set bits in integer.
(c) Write a program to check if number is power of two without using loops.
(d) Write a program to find gcd and lcm of two positive integers.
3. (a) Write a program to find sin(x), cos(x), ex using series.
(b) Write a program to find an suing squaring method (use bitwise operators).
(c) Write a program to find first n Fibonacci numbers.
4. (a) Write function to check if given integer is prime or not and use it find all primes ingiven range.
(b) Given unsorted list of n-1 distinct integers where integers are in the range 1 to n, findmissing number using XOR.
(c) Given unsorted list of n-2 distinct integers where integers are in the range 1 to n, findtwo missing numbers using XOR.
5. String manipulations using pointers.
(a) String length
(b) Display substring from a given position and up to the given number of characters
(c) Concatenate two strings
(d) Uppercase to Lowercase.
6. (a) String compare Without using Standard Library functions.
(b) Test whether string is palindrome.
(c) Convert number to string and string to number.
(d) Convert a number to any base (2 to 36).
7. Matrix operations.
(a) Write a program to find addition, subtraction, multiplication, transpose of matrix.
(b) Write a program to interchange rows, columns of matrix.
(c) Write a program to construct a magic square of odd size using array of pointers.
8. (a) Write a program to remove duplicate numbers in sorted array using pointer.
(b) Define a variable argument function like printf.
9. Write a program to create a student record using struct, create list of student records,support the operations to add new record, delete some record etc.
10. Define Complexnumbers and rationals using struct. Define functions to add, subtract numbers.
11. Write a program to count number of chars, words, vowels, digits, in a text file. Usecommand line argument to supply file name.
12. Write a program to create binary file to store students record. Define functions to search, add, delete records in file.

Semester - II

MIM 201: Complex Analysis

Unit 1. Complex Numbers **[08 hours]**

1.1 : Sums and Products
1.2 : Basic Algebraic Properties
1.3 : Further Properties
1.4: Vectors and Moduli
1.5: Complex Conjugates
1.6: Exponential Form
1.7: Products and Powers in Exponential Form
1.8: Arguments of Products and Quotients
1.9:Roots of Complex Numbers
1.10: Examples
1.11: Regions in the Complex Plane.

Unit 2. Analytic functions **[12 hours]**

2.1 : Functions of Complex Variables
2.2 : Mappings
2.3 :Limits
2.4 :Theorems on Limits
2.5 :Limits Involving the Point at Infinity
2.6 : Continuity
2.7 : Derivatives
2.8 :Differentiation Formulas
2.9 : Cauchy- Riemann Equations
2.10 :Sufficient Conditions for Differentiability
2.11 :Polar Coordinates
2.12 :Analytic Functions
2.13 : Examples
2.14 :Harmonic Functions
2.15 :Uniquely Determined Analytic Functions.

Unit 3. Elementary Functions **[08 hours]**

3.1 : The Exponential Functions
3.2 : The Logarithmic Function
3.3 : Branches and Derivatives of Logarithms
3.4 : Some Identities Involving Logarithms
3.5 : Complex Exponents
3.6 : Trigonometric Functions
3.7 :Hyperbolic Functions
3.8 :Inverse Trigonometric and Hyperbolic Functions.

Unit 4. Integrals **[12 hours]**

4.1 : Derivatives of Functions
4.2 : Definite Integrals of Functions
4.3 : Contours
4.4 :Contour Integral
4.5 : Some Examples
4.6 :Examples with Branch Cuts

4.7 : Upper Bounds for Moduli of Contour Integrals
4.8 : Anti-Derivatives
4.9 : Proof of the Theorem
4.10 :Cauchy-Goursat's Theorem (without proof)
4.11 :Simply and Multiply Collected Domains (only definitions)
4.12 : Cauchy Integral Formula
4.13 : An Extension of Cauchy Integral Formula
4.14 : Some Consequences of the Extension(Morera Theorem)
4.15 : Derivatives of Analytic Functions
4.16 :Liouville's Theorem and Fundamental Theorem of Algebra
4.17 : Maximum Modulus Principle

Unit 5. Series **[08 hours]**
5.1 :Convergence of Sequence
5.2 :Convergence of Series
5.3 : Taylor's Series
5.4 : Proof of Taylor's Theorem
5.5 : Examples
5.6 :Laurent Series
5.7 :Proof of Laurent's theorem
5.8 : Examples

Unit 6. Residues and Poles **[08 hours]**
6.1 : Isolated Singular Points
6.2 : Residues
6.3 : Cauchy Residue Theorem
6.4 : Residue at Infinity
6.5 :The three Types of Isolated Singular Points
6.6 : Residues at poles
6.7 : Zeros of Analytic Functions
6.8 : Zeros and Poles
6.9 : Behavior of Functions Near Isolated Singular Points

Unit 7. Applications of Residues **[04 hours]**
7.1 :Evaluation of Improper Integrals
7.2 :Examples

Text Book:
J.W. Brown and R.V. Churchill, Complex Variables and Applications, International Student Edition, 2009.(Eighth Edition, Tata McGraw Hill).
Chapter1 : Sec. 1 to 11.
Chapter 2: Sec.12 to 13, Sec. 15 to 27.
Chapter 3 : Sec. 29 to 36.
Chapter 4 : Sec. 37 to 46 and Sec. 48 to 54.
Chapter 5 : Sec. 55 to 62.
Chapter 6: Sec. 68 to 77.
Chapter 7: Sec. 78 to 79.

Reference Books:
1. S. Ponnusamy, Complex Analysis, Second Edition (NarosaPublishing house).
2. John B. Conway : Functions of one complex variable (Narosa Publishing house)

3. S. Lang, Complex Analysis, (Springer Verlag).
4. A.R. Shastri, An Introduction to Complex Analysis, (MacMillan).
5. Joseph Bak, Donald F. Newman, Complex Analysis (Second Edition, Springer).

MIM 202: Discrete Mathematical Structures

Unit 1: Counting Techniques [16 Lectures]

1.1 The Basics of Counting
1.2 The Pigeonhole Principle
1.3 Permutations and Combinations
1.4 Binomial Coefficients
1.5 Generalized Permutations and Combinations
1.6 Applications of Recurrence Relations
1.7 Solving Linear Recurrence Relations
1.8 Divide-and-Conquer Algorithms and Recurrence Relations

Unit 2: Graphs [16 Lectures]

2.1 Graphs and Graph Models
2.2 Graph Terminology and Special Types of Graphs
2.3 Representing Graphs and Graph Isomorphism
2.4 Connectivity
2.5 Euler and Hamilton Paths
2.6 Shortest-Path Problems
2.7 Planar Graphs
2.8 Graph Coloring

Unit 3: Trees [16 Lectures]

3.1 Introduction to Trees
3.2 Applications of Trees
3.3 Tree Traversal
3.4 Spanning Trees
3.5 Minimum Spanning Trees

Unit 4: Boolean Algebra [12 Lectures]

4.1 Boolean Functions
4.2 Representing Boolean Functions
4.3 Logic Gates
4.4 Minimization of Circuits

Text Book:

Kenneth H. Rosen, DISCRETE MATHEMATICS AND ITS APPLICATIONS, SEVENTH EDITION, Tata McGrew-Hill Publishing Company Limited, Special Indian Edition 2007 (6.1-6.5, 8.1-8.3, 10.1-10.8, 11.1-11.5, 12.1-12.4)

Reference Books:

1. Narsingh Deo, Graph Theory with Applications to Computer Science and

Engineering, PHI Publication.

3. C. L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill.

MIM 203: Data Structures

Course Objectives:

1. To teach the concept of fundamental data structures, for data storage and fast retrieval
2. Focus on Efficient algorithms, with respect to different storage methods
3. To Provide a solid background in algorithms, from the Computer science perspective.
4. To understand the pros and cons of various sorting and searching methods

Course Outcomes:

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

1. To discriminate on the usage of various data structures , in deriving a problem solution.
2. To design algorithms to solve programming problems
3. To use effective and appropriate data structures in solving Computer Science domain problems
4. To use appropriate algorithmic strategy for better efficiency.

Unit 1: Introduction to data structures **[2 Lectures]**

1.1 Concept
1.2 Data type, Data object
1.3 ADT
1.4 Need of Data Structure
1.5 Types of Data Structure

Unit 2: Algorithm analysis **[6 Lectures]**

2.1 Algorithm – definition, characteristics

2.2 Goal of analysis of algorithms

2.3 Running time analysis

2.4 How to compare algorithms

2.5 Rate of growth, Commonly used rates of growth

2.6 Types of analysis

2.7 Space complexity, time complexity

2.8 Asymptotic notation (Big O, Omega _)

Unit 3: Linear data structures **[10 Lectures]**

3.1 Introduction to Arrays - array representation

3.2 Sorting algorithms with efficiency : Comparison based(

Bubble sort, Quick sort, Insertion sort , Merge sort), Non-Comparison based (Counting sort, Radix sort, Bucket sort)

3.3 Comparison of Sorting algorithms.

Unit 4: Linked List [8 Lectures]

4.1 Introduction to List
4.2 Implementation of List – static & dynamic representation
4.3 Types of Linked List : singly, Doubly, Circular
Unrolled Link lists, Skip lists
4.4 Operations on List
4.5 Applications of Linked List – polynomial manipulation
4.6 Generalized linked list – concept & representation

Unit 5: Stacks [6 Lectures]

5.1 Introduction
5.2 Representation-static & dynamic
5.3 Operations
5.4 Application - infix to postfix & prefix, postfix evaluation,
5.5 Recursion using implicit stack
5.6 Concept of Multiple stacks

Unit 6: Queues [8 Lectures]

6.1 Introduction
6.2 Representation -static & dynamic
6.3 Operations
6.4 Circuler queue, DeQue, priority queues
6.5 Concept of Multiple Queues

Unit 7: Trees [14 Lectures]

7.1 Concept & Terminologies
7.2 Binary tree, binary search tree
7.3 Representation – static & dynamic
7.4 Operations on BST – create. Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes
7.5 Threaded binary tree traversals,
7.6 XOR trees, Expression trees
7.7 Application - Heap sort
7.8 Height balance tree- AVL trees- Rotations

Unit 8: Graph [6 Lectures]

8.1 Concept & terminologies
8.2 Graph Representation
8.3 Traversals – BFS & DFS
8.4 Applications – AOV network – topological sort
8.5 AOE network – critical path
8.6 Shortest path with implementation

References:

1. Fundamentals of Data Structures ---- By Horowitz Sahani (Galgotia)
2. Data Structure and Algorithms made easy: By NarasimhaKarumanchi
3. Data Structures using C --- By ISRD Group (Tata McGraw Hill)
4. Introduction to Data Structures using C---By Ashok Kamthane
5. Data Structures using C --- Bandopadhyay&Dey (Pearson)

MIM 204: Software Engineering

Course Objectives:

1. To learn and understand the principles of Software Engineering
2. To be acquainted with methods of capturing, specifying, visualizing and analyzing software requirements.
3. To understand O-O S/W development life cycle, Using O.O.S.E principles
4. To apply a Modeling Language (UML) for modeling an Object oriented S/W system development.
5. To understand the concept of testing an O-O S/W.

Course Outcomes:

On completion of the course, student will be able to–

1. Designvarious modelsto depict the development of a object oriented software system.
2. Design test cases of a software system.

Unit 1 : Introduction to Software Engineering, Software Process models **[10 Lectures]**

1.1: Nature of Software
1.2: Software Engineering Principles
1.3: The Software Process
1.4: Software Myths.
1.5: Process Models **:**A Generic Process Model
1.6: Prescriptive Process Models: The Waterfall, Incremental Process (RAD), Evolutionary Process, Unified Process, Concurrent.
1.7: Advanced Process Models & Tools**:** Agile software development: Agile methods, Plan-driven and agile development, Extreme programming Practices, Testing in XP, Pair programming.
1.8: Introduction to agile tools: JIRA, Kanban,

Unit 2: Object Oriented Concepts and Principles **[6 Lectures]**

2.1: What is Object Orientation
2.2: Introduction to an Object, Classes and Instance
2.3: Polymorphism

2.4: Inheritance
2.5: Object Oriented System Development
2.5.1: Introduction
2.5.2: Function/Data Methods (With Visibility)
2.5.3: Object Oriented Analysis
2.5.4: Object Oriented Construction
Identifying the Elements of an Object Model,
Identifying Classes and Objects, Specifying the
Attributes (With Visibility) , Defining Operations
Finalizing the Object Definition

Unit 3: Introduction to UML **[2 Lectures]**

3.1: Concept of UML
3.2: Advantages of UML

Unit 4: **Basic Structural Modeling** **[5 Lectures]**

4.1: Classes
4.2: Relationships
4.3: Common Mechanisms
4.4: Class Diagram

Unit 5: Advanced Structural Modeling **[7 Lectures]**

5.1: Advanced Classes
5.2: Advanced Relationship
5.3: Interface, Types and Roles
5.4: Packages
5.5: Object Diagrams

Unit 6 : Basic BehavioralModeling **[9 Lectures]**

6.1: Interactions
6.2: Use Cases and Use Case Diagram with stereo types
6.3: Interaction Diagram
6.4: Sequence Diagram
6.5: Activity Diagram
6.6: State Chart Diagram

Unit 7: Object Oriented Analysis **[6 Lectures]**

7.1: Iterative Development and the Rational Unified Process
7.2: Inception
7.3: Understanding Requirements
7.4: Use Case Model From Inception to Elaboration
7.5: Elaboration

Unit 8: Object Oriented Design **[4 Lectures]**

8.1: The Generic Components of the OO Design Model
8.2: The System Design Process
8.2.1: Partitioning the Analysis Model
8.2.2: Concurrency and Sub System Allocation
8.2.3: Task Management Component
8.2.4: The Data Management Component
8.2.5: The Resource Management Component
8.2.6: Inter Sub System Communication
8.3: Object Design Process

Unit 9: Architectural Modeling **[6 Lectures]**

9.1:Component
9.2: Components Diagram
9.3: Artifacts of the system &Artifact diagram
9.4 : Deployment Diagram
9.5: Collaboration Diagram

Unit 10 : Object Oriented Testing **[5 Lectures]**

10.1: Object Oriented Testing Strategies
10.2: Test Case Design for Object Oriented Software
10.3: Inter Class Test Case Design

References:

1. Roger Pressman, ―Software Engineering: A Practitioner‘s Approach‖, McGraw Hill, ISBN 0–07–337597–7
2. Ian Sommerville, ― Software Engineering‖, Addison and Wesley, ISBN 0-13-703515-
3. Grady Booch, James Rambaugh,"The Unified Modeling Language User/Reference Guide",Pearson Education INC
4. Ivar Jacobson, "Object Oriented Software Engineering", Pearson Education INC
5. Craig Larman, "Applying UML and Patterns", Pearson Education INC
6. Bennett, Simon," Object Oriented Analysis and Design" McGraw Hill

MIM 205 JAVA

Unit 1. Introduction to Object Oriented Concepts [05 lectures]

1.1. Object, Class
1.2. Encapsulation, Abstraction, Data Hiding, Inheritance, Polymorphism
1.3. Message Passing, Dynamic binding
1.4. History of Object Oriented languages
1.5. Comparison with structured programming

Unit 2. Introduction to The Java Technology [04 lectures]

2.1. The Java platform, Java buzzwords, API, JVM
2.2. Java compiler, bytecodes
2.3. java editions

Unit 3. Main features of Java language [06 lectures]

3.1. Introduction to Java, Writing & compiling Java programs- the main method
3.2. Command line arguments, String class, Primitive data types, Variables and assignment, javadoc comments
3.3. Expressions, Data conversion, Interactive programs, Boolean data type and expressions { If, Switch } statements, { For, While, Do } statements, Creating, callingmethods, Parameter passing, Returning values, Overloading methods, Scope of variables.

Unit 4. Arrays [05 lectures]

4.1. Defining and initializing arrays, new operator, using arrays
4.2. passing arrays to methods, returning arrays from methods
4.3. command-line arguments
4.4. 2-dimensional arrays

Unit 5. Objects and Classes [05 lectures]

5.1. Defining Class, Creating object, reference variables
5.2. Visibility modifiers – public, private, protected
5.3. Object members and class members (static), Arrays of objects, this keyword, Wrapper Classes

Unit 6. Packages and Interfaces [05 lectures]

6.1. Concept of package, Package and import keywords
6.2. Concept of interfaces, Implementing interfaces
6.3. Use of predefined packages
6.4. Use of predefined interfaces – Comparable and Comparator

Unit 7. Inheritance and Polymorphism [05 lectures]

7.1. Superclass and Subclass – extends keyword, super keyword, Overriding members
7.2. Protected data members- Object Class and its toString() method,Abstract Classes
7.3. Final classes, methods and variables, instanceof operator
7.4. dynamic binding , Casting objects

Unit 8. Exceptions and Exception handling [05 lectures]

8.1. Exception class hierarchy
8.2. Checked and unchecked exceptions
8.3. Try, catch, throw, throws finally keywords

8.4. Creating user defined exceptions

Unit 9. Text and File I/O [05 lectures]

9.1. Predefined I/O classes
9.2. Simple I/O operations using console and files
9.3. The File class

Unit 10. GUI and Event Handling using Java [05 lectures]

10.1. Introduction to AWT and Swing
10.2. Creating containers and components (JFrame, JPanel, JButton, JTextField, JCheckBox, JRadioButton, JMenu, JList, JTable)
10.3. Layout Managers
10.4. Delegation event model - Event sources, event listeners, event classes.

Unit 11. JDBC [05 lectures]

11.1. The Design of JDBC
11.2. The Structured query language
11.3. Basic JDBC programming concepts
11.4. Query Execution
11.5. Scrollable and updatable result sets.

Unit 12. Introduction to collections [05 lectures]

12.1. Concrete Collections
12.1.1. Linked List
12.1.2. Array Lists
12.1.3. Hash Sets
12.1.4. Tree Sets
12.1.5. Maps

Reference Books:

1. Java : How to Program, Deitel&Deitel, Prentice Hall
2. Core Java 2: Volume I – Fundamentals, Cay S. Horstmann and Gary Cornell; Prentice-Hall 2002. ISBN 0130471771
3. Core Java 2: Volume II – Advanced Features, Cay S. Horstmann and Gary Cornell; Prentice-Hall 2001. ISBN 0130927384
4. Java: The Complete Reference, Herbert Schildt. Fifth Edition
5. Introduction to Java Programming, Daniel Liang

Important URLs :http://java.sun.com/reference/docs/

MIM – 206 : Lab Work on Data Structures

1. Assignment on Sorting / Searching algorithms, using Linear data structures(Arrays), with algorithm analysis for complexity.
2. Assignment on Linked List (Creation/updation on singly/doubly/circular/unrolled/skip lists)
3. Assignment on Creation and application of stack (static/dynamic implementation)
4. Assignment on Creation and application of Queue (static/dynamic implementation)
5. Assignment on Creation and application of Trees (Binary/AVL implementation)
6. Assignment on Creation and application of Graphs