Progressive Education Society's

Modern college of Arts, Science and Commerce,

Ganeshkhind,Pune-16

Autonomous

(Under Faculty of Science and Technology)

S.Y.B.Sc.(Computer Science)) : Mathematics

Choice Based Credit System Syllabus

To be implemented from Academic Year 2023-2024

S. Y. B. Sc. (Computer Science)

Introduction: Taking into consideration the rapid changes in science and technology and new approaches in different areas of mathematics and related subjects Board of studies in Mathematics of Modern College , Ganeshkhind , Pune has prepared the syllabus of S.Y.B.Sc. Computer Science Mathematics. To develop the syllabus the U.G.C. Model curriculum is followed.

Aims:

i) Give the students a sufficient knowledge of fundamental principles ,methods and a clear perception of innumerous power of mathematical ideas and tools and know 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 skills , 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 as notations, 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, mathematical reasoning.

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

(iv) A student 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.

Structure of S. Y. B. Sc. Mathematics (Computer Science) Course:

	Semester - III		Semester -IV		Credits
Paper I	**23-MT-231**	**Groups and Coding Theory**	**23-MT-241**	**Computational Geometry**	**2**
Paper II	**23-MT-232**	**Numerical Analysis**	**23-MT-242**	**Operations Research**	**2**
Paper III	**23-MT-233**	**Mathematics Practical**	**23-MT-243**	**Mathematics Practical**	**2**

Medium of Instruction: English

*** Eligibility:** F.Y.B.Sc. Computer Science, as per University rules.

Examination:

A) **Pattern of Examination:** Semester

B) Each course is of 50 marks (35 marks theory & 15 marks internal examination).

C) **Standard of passing :** 20 marks out of 50 marks for each paper. But the student should obtain minimum 14 marks out of 35 in the theory examination and 6 marks for internal examination out 0f 15.

D) Pattern of Question Papers for papers 23-MT-231, 23-MT-232, 23-MT-241, 23-MT-242.

Q.1 Attempt any 5 out of 7 each of 2 marks [10 marks].

Q.2 Attempt any 3 out of 5 each of 5 marks [15 marks].

Q.3 Attempt any 1 out of 2 each of 10 marks [10 marks].

E) **External students:** Not Allowed

Detailed Syllabus

Semester – III
23-MT-231 : Groups and Coding Theory
(2 Credits)

Course Learning Outcomes

CO1: Student will get introduced to the fundamental theory of groups.

CO2: Student will learn the concepts of congruence relation and equivalence class partition

CO3: Student will learn in detail the examples of finite groups such as Zn and Permutation group

CO4: Student will be able to explain the significance of the notion of cosets and its application in Coding theory

CO5: Student will learn Public Key Cryptography for data security.

Course Contents

1. **Integers** **[06 Lectures]**
   1.1 Division Algorithm
   1.2 GCD using division algorithm and expressing it as a linear combination
   1.3 Euclid's Lemma
   1.4 Equivalence relation (revision), Congruence relation on set of integers, Equivalence class partition.
2. **Groups** **[06 Lectures]**
   2.1 Binary Operation
   2.2 Group: Definition and Examples
   2.3 Elementary Properties of Groups
3. **Finite Groups and Subgroups** **[14 Lectures]**
   3.1 Order of a group, order of an element
   3.2 Examples of $(Zn,+)$ and $(U(n),*)$
   3.3 Subgroup definition, finite subgroup test, subgroups of Zn
   3.4 Generator of a group, cyclic group, finding generators of Zn (Corollary 3,4 without Proof)
   3.5 Permutation group, definition, composition of two permutations, representation as a product of disjoint cycles, inverse and order of a permutation, even/odd permutation
   3.6 Cosets: definition, Examples and properties, Lagrange Theorem(Without Proof)
4. **Groups and Coding Theory** **[10 Lectures]**
   4.1 Coding of Binary information and Error detection
   4.2 Decoding and Error correction
   4.3 Public Key Cryptography

Text Books:

1. Contemporary Abstract Algebra By J.A. Gallian (Seventh Edition)
   Unit 1: Chapter 0, Unit 2 : Chapter 2, Unit 3 : Chapter 3,4,5 and 7
2. Discrete Mathematical Structures By Bernard Kilman, Robert C. Busby and Sharon Ross (6^{th} Edition) Pearson Education Publication
   Unit 4 : Chapter 11

Reference Books :

1. A First Course in Abstract Algebra, By J. B. Fraleigh, Pearson.
2. Topics in Algebra By I. N. Herstein, John Wiley and Sons.

23-MT-232 : Numerical Analysis

(2 Credits)

Course Learning Outcomes

CO1: Student will learn to apply the various numerical techniques for solving real life problems.

CO2: Student will be able to solve the integration problems which cannot be solved by usual formulae and methods using numerical techniques.

CO3: Student will learn curve fitting to the data using 3 different methods of interpolation

CO4: Student will be able to find approximate solutions to differential equations occurring in engineering sciences.

Course Contents

1. Algebraic and Transcendental Equation [04 Lectures]

1.1 Introduction to Errors

1.2 False Position Method

1.3 Newton-Raphson Method

2. Calculus of Finite Differences and Interpolation [16 Lectures]

2.1 Differences

2.1.1 Forward Differences

2.1.2 Backward Differences

2.1.3 Central Differences

2.1.4 Other Differences (δ, μ operators)

2.2 Properties of Operators

2.2.1 Relation between Operators

2.2.2 Newton's Gregory Formula for Forward Interpolation

2.2.3 Newton's Gregory Formula for Backward Interpolation

2.2.4 Lagrange's Interpolation Formula

2.2.5 Divided Difference

2.2.6 Newton's Divided Difference Formula

3 Numerical Integration [08 Lectures]

3.1 General Quadrature Formula

3.2 Trapezoidal Rule

3.3 Simpson's one-Third Rule

3.4 Simpson's Three-Eight Rule

4 Numerical Solution of Ordinary Differential Equation [08 Lectures]

4.1 Euler's Method
4.2 Euler's Modified Method
4.3 Runge-Kutta Methods

Text Book:-

A textbook of Computer Based Numerical and Statistical Techniques, by A. K. Jaiswal and Anju Khandelwal. New Age International Publishers. Chapter 1:2.1, 2.5, 2.7 Chapter 2:3.1, 3.2, 3.4, 3.5,4.1, 4.2, 4.3, 5.1, 5.2, 5.4, 5.5 Chapter 3:6.1, 6.3, 6.4, 6.5, 6.6, 6.7 Chapter 4:7.1, 7.4, 7.5, 7.6

Reference Books:-

1. S.S. Sastry; Introductory Methods of Numerical Analysis, 3rd edition, Prentice Hall of India, 1999.
2. H.C. Saxena; Finite differences and Numerical Analysis, S. Chand and Company.
3. K.E. Atkinson; An Introduction to Numerical Analysis, Wiley Publications.
4. Balguruswamy; Numerical Analysis.

23 MT-233 Python Programming Language-I

(2 Credits)

Course Learning Outcomes

CO1: Student will understand how Python is useful scripting language for developers.

CO2: Student will be able to use lists, tuples & dictionaries in Python programs.

CO3: Student will acquire programming skills in core Python.

CO4: Student will learn to solve Linear Algebra problems using Python.

CO5: Student will learn to write program for Numerical methods.

Course Contents

1. Introduction to Python

1.1. Installation of Python
1.2. Values and types: int, float and str,
1.3. Variables: assignment statements, printing variable values, types of variables.
1.4. Operators, operands and precedence:+, -, /, *, **, % PEMDAS(Rules of precedence)
1.5. String operations: + : Concatenation, * : Repetition
1.6. Boolean operator:
  1.6.1.Comparison operators: ==, !=, >, =, <=
  1.6.2.Logical operators: and, or, not
1.7. Mathematical functions from math, cmath modules.
1.8. Keyboard input: input() statement

2. **String, list, tuple**
   - 2.1. Strings:
     - 2.1.1. Length (Len function)
     - 2.1.2. String traversal: Using while statement, Using for statement
     - 2.1.3. String slice
     - 2.1.4. Comparison operators (>, <, ==)
   - 2.2. Lists:
     - 2.2.1. List operations
     - 2.2.2. Use of range function
     - 2.2.3. Accessing list elements
     - 2.2.4. List membership and for loop
     - 2.2.5. List operations
     - 2.2.6. Updating list: addition, removal or updating of elements of a list
   - 2.3. Tuples:
     - 2.3.1. Defining a tuple,
     - 2.3.2. Index operator,
     - 2.3.3. Slice operator,
     - 2.3.4. Tuple assignment,
     - 2.3.5. Tuple as a return value
3. **Iterations and Conditional statements**
   - 3.1. Conditional and alternative statements, Chained and Nested Conditionals: if, if-else, if-elif-else, nested if, nested if-else
   - 3.2. Looping statements such as while, for etc, Tables using while.
   - 3.3. Functions:
     - 3.3.1. Calling functions: type, id
     - 3.3.2. Type conversion: int, float, str
     - 3.3.3. Composition of functions
     - 3.3.4. User defined functions, Parameters and arguments

4. **Linear Algebra**
   - 4.1. Matrix construct, eye(n), zeros(n,m) matrices
   - 4.2. Addition, Subtraction, Multiplication of matrices, powers and invers of a matrix.
   - 4.3. Accessing Rows and Columns, Deleting and Inserting Rows and Columns
   - 4.4. Determinant, reduced row echelon form, nullspace, columnspace, rank
   - 4.5. Eigenvalues, Eigenvectors, and Diagonalization
5. **Numerical methods in Python**
   - **5.1.** Simple Iterations Method
     - 5.1.1 Bisection Method
     - 5.1.2 Newton-Raphson Method
     - 5.1.3 False Position (Regula Falsi) Mehtod
   - 5.2 Numerical Integration:
     - 5.2.1 Trapezoidal Rule,
     - 5.2.2 Simpson's 1/3 Rule,
     - 5.2.3 Simpson's 3/8 Rule

Text Books:-

1. Downey, A. et al., How to think like a Computer Scientist: Learning with Python, John Wiley, 2015.
   Sections: 1, 2, 3
2. Robert Johansson, Introduction to Scientific Computing in Python Section: 4

Reference Books:-

1. Lambert K. A., Fundamentals of Python - First Programs, Cengage Learning India, 2015.
2. Guzdial, M. J., Introduction to Computing and Programming in Python, Pearson India.
3. Perkovic, L., Introduction to Computing Using Python, 2/e, John Wiley, 2015.
4. Zelle, J., Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates Inc.
5. Sandro Tosi, Matplotlib for Python Developers, Packt Publishing Ltd.(2009)

List of Practicals

Practical 1: Introduction to Python, Python data type-I (Unit 1)

Practical 2: Python data type-II (Unit 2)

Practical 3: Iterations & conditional statements (Unit-3 – 3.1,3.2)

Practical 4: Iterations & conditional statements (Unit 3 – 3.3)

Practical 5: Application : Matrices (Unit 4 – 4.1,4.2,4.3)

Practical 6: Application : Matrices (Unit 4 – 4.4,4.5)

Practical 7: Application : System of equations (Unit 4 – 4.5)

Practical 8: Application : Eigenvalues ,Eigenvectors (Unit 4 – 4.6)

Practical 9: Application : Roots of equations (Unit 5 – 5.1)

Practical 10: Application: Numerical Integration. (Unit 5-5.2)

Practical 11: (Written Practical) (Unit 4 of MTC-231) Groups & Coding Theory.

Practical 12: (Written Practical) (Unit 4 of MTC-232) Numerical Solutions of ODE.

Practicals 1 to 10 are Python Programming Practicals.

23 MT-241: Computational Geometry
(2 Credits)

Course Learning Outcomes

CO1: Student will gain detail knowledge of the fundamental concepts in Computational

Geometry

CO2: Student will learn general tachniques for solving problems in Computational

Geometry

CO3: Student will be able to characterize invariance properties of Euclidean geometry by

groups of transformation.

CO4: Student will be able to describe & construct basic geometric shapes such as circle

& parabola by computational means.

CO5: Student will learn Be'zier curve fitting.

Course Contents

1. Two dimensional transformations: [12 Lectures]

1.1 Introduction.
1.2 Representation of points.
1.3 Transformations and matrices.
1.4 Transformation of points.
1.5 Transformation of straight lines
1.6 Midpoint Transformation
1.7 Transformation of parallel lines
1.8 Transformation of intersecting lines
1.5 Transformation: rotations, reflections, scaling, shearing.
1.6 Combined transformations.
1.7 Transformation of a unit square.
1.8 Solid body transformations.
1.9 Translations and homogeneous coordinates.
1.10 Rotation about an arbitrary point.
1.11 Reflection through an arbitrary line.

2. Three dimensional transformations: [08 Lectures]

2.1 Introduction.
2.2Three dimensional – Scaling, shearing, rotation, reflection, translation.
2.3 Multiple transformations.
2.4 Rotation about – an axis parallel to coordinate axes, an arbitrary line
2.5 Reflection through – coordinate planes, planes parallel to coordinate planes , an arbitrary plane

3. Projection [08 Lectures]

3.1 Orthographic projections.
3.2 Axonometric projections.
3.3 Oblique projections
3.4 Single point perspective projection

4. Plane and space Curves: [08 Lectures]

4.1 Introduction.
4.2 Curve representation.
4.3 Parametric curves.
4.4 Parametric representation of a circle and generation of circle.
4.5 Parametric representation of a parabola and generation of arc of parabola.
4.6 Bezier Curves – Introduction, definition, properties (without proof), Curve fitting (up to n = 3), equation of the curve in matrix form (upto n = 3)

Textbook: D. F. Rogers, J. A. Adams, Mathematical elements for Computer graphics,

Mc Graw Hill Intnl Edition. Chapter 1: 2.1 to 2.17; Chapter 2: 3.1 to 3.10, Chapter 3: 3.12 to 3.14; Chapter 4: 4.1, 4.2, 4.5, 4.7, 5.1, 5.8

Reference books:

- Computer Graphics with OpenGL, Donald Hearn, M. Pauline Baker, Warren Carithers, Pearson (4th Edition)
- Schaum Series, Computer Graphics.

23 MTC-242: Operations Research
(2 Credits)

Course Learning Outcomes

CO1: Student will be able to analyze & solve LPP models.

CO2: Student will learn the graphical solution of LPP with only two variables.

CO3: Student will learn the simplex method for solving LPP

CO4: Student will learn the real life applications such as Transportation & Assignment Problem.

Course Contents

1. **Linear Programming Problem I** **[12 Lectures]**
   - 1.1 Introduction Definition and Examples
   - 1.2 Problem solving using Graphical method
   - 1.3 Theory of Linear Programming, Slack and surplus variables, Standard form of LPP,
     Some important definitions, Assumptions in LPP, Limitations of Linear programming, Applications of Linear programming, Advantages of Linear programming Techniques
   - 1.4 Simplex method, Big- M-method
2. **Linear Programming Problem II** **[08 Lectures]**
   - 2.1 Special cases of LPP : Alternative solution, Unbounded solution, Infeasible solution
   - 2.2 Duality in Linear Programming, Primal to dual conversion, Examples
3. **Assignment Models** **[06 Lectures]**
   - 3.1 Assignmment Model -Introduction
   - 3.2 Hungerian method for Assignment problem
4. **Transportation Models** **[10 Lectures]**
   - 4.1 Introduction, Tabular representation
   - 4.2 Methods of IBFS (North-West rule, Matrix-minima, Vogel's Approximation), Algorithms
   - 4.3 The Optimality Test of Transportation Model (MODI method only)

Text Book:-Operation Research (12 th Edition), by S.D.Sharma.

Chapter 1: 1.1,1.3-1,1.3-2,1.5,1.6,1.8,1.9,1.10,1.11,1.12,3.1,3.2,3.3,3.4,3.5-4,
Chapter 2: 3.8-1,3.8-2,5.1-1, 5.2-1,5.3,5.7-1, 5.7-2
Chapter 3: 9.1, 9.2, 9.4-1, 9.4-2, 9.5, 9.6, 9.7-1, 9.7-2
Chapter 4: 10.1, 10.2, 10.5, 10.8-1,10.9, 10.10

Reference Books:-

1. Operations Research by H. A. Taha
2.Operations Research by R. Panneerselvam, Prentice Hall of India.
3. Principles of Operations Research by H. M. Wagner, Prentice Hall of India.
4. Operations Research by Gupta and Hira.
5. Operation Research by J.K. Sharma

23 MT-243 Python Programming Language-II

(Credits)

Course Learning Outcomes

CO1: Student will acquire object oriented skills in Python.

CO2: Student will be able to demonstrate the use of Python in Numerical Integration &

Computational Geometry.

CO3: Student will study graphics & design & implement a program to solve a real world

Problem.

CO4: Student will be able to solve Linear Programming Problem using Python.

Course Contents

1. **2D, 3D Graphs**
   - 1.1. Installation of numpy, matplotlib packages
   - 1.2. Graphs plotting of functions such as x^2, x^3, $sine$, $cosine$, log, e^x etc.
   - 1.3. Different formats of graphs.
   - 1.4. Three-dimensional Points and Lines
   - 1.5. Three-dimensional Contour Plots
   - 1.6. Wireframes and Surface Plots
   - 1.7. Graphs plotting of functions such as $x^2 + y^2$, $sin(x + y)$, $log(x + y)$, $e^{x^2+y^2}$ etc.
2. **Computational Geometry**
   - **2.1.** The distance between two points, List of points- the point List class, Integer point lists, ordered point sets, Extreme points of a PointList, Random sets of Points not in general position
   - **2.2.** Displaying Points and other geometrical objects, Lines, rays and line segments, the geometry of line segments.
   - **2.3.** Polygon: Representing polygons in python, Triangles, signed area of a triangle, Triangles and the relationships of points to lines, is collinear, is Left, is Left On, is Right, is Right On, Between.
   - **2.4.** Two dimensional rotation and reflection.

2.5. Three dimensional rotation and reflection

3. Study of Operational Research in Python

3.1 Linear Programming in Python

3.2 Simplex Method in Python

Text Books:-

1. Jaan Kiusalaas, Numerical Methods in Engineering with Python, Cambridge University Press,(2005)
   Sections: 3
3. Robert Johansson, Introduction to Scientific Computing in Python
   Section: 1
2. Jason Brownlee, Basics of Linear Algebra for Machine Learning, Discover the Mathematical Language of Data in Python
   Sections: 2

Reference Books:-

1. Lambert K. A., Fundamentals of Python - First Programs, Cengage Learning India, 2015.
2. Guzdial, M. J., Introduction to Computing and Programming in Python, Pearson India.
3. Perkovic, L., Introduction to Computing Using Python, 2/e, John Wiley, 2015.
4. Zelle, J., Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates Inc.
5. Jim Arlow, Interactive Computational Geometry in Python.

List of Practicals

Practical 1: Graph plotting (Unit 1- 1.1 to 1.3)

Practical 2: Graph plotting (Unit 1- 1.4 to 1.7)

Practical 3: Application to Computational Geometry (Unit 2 – 2.1,2.2)

Practical 4: Application to Computational Geometry (Unit 2 – 2.3)

Practical 5: Application to Computational Geometry (Unit 2 – 2.4,2.5)

Practical 6: Study of effect of concatenation of two dimensional & three dimensional transformation.

Practical 7: Study of Operation Research in Python (Unit 3 – 3.1)

Practical 8: Study of Operation Research in Python (Unit 3 – 3.2)

Practical 9: Study of graphical aspects of two dimensional transformation matrix using matplotlib.

Practical 10: (Written Practical) (Unit 4 of MTC-241) Plane & Space curve.

Practical 11: (Written Practical) (Unit 3 of MTC-242) Assignment Models.

Practical 12: (Written Practical) (Unit 4 of MTC-242) Transportation Models.

Practical 1 to 9 are to be done using Python Programming.

Modalities for conducting practicals and practical Examination :

1. There will be one 4 hours & 20 minutes (260 minutes) practical session for each of batch of 15 students per week for each practical course.
2. External examiner shall be appointed by the college for Mathematics Practical Examination.
3. The duration of practical examination is 3 hours.
4. The practical examination is of 35 marks which consist of written examination of 10 marks & 25 marks on Python programming. The slips for the questions on programming and problem solving shall be prepared by the examiner.
5. The internal 15 marks will be given on the basis of journal prepared by student and the cumulative performance of student at practical .
6. Study tours may be arranged at place having important mathematical institutes or historical places.