Anekant Education Society's

Tuljaram Chaturchand College of Arts, Science and Commerce, Baramati

(Autonomous)

Course Structure for B.Sc. (Computer Science) Mathematics (2022 Pattern)

F. Y. B. Sc. (Computer Science) Mathematics

Semester	Course Code	Title of Course	No. of Credits	No. of Lectures
I	UCSMT111	Graph Theory	2	36
	UCSMT112	Matrix Algebra	2	36
	UCSMT113	Mathematics Practical based on UCSMT111 & UCSMT112	2	48
II	UCSMT121	Discrete Mathematics	2	36
	UCSMT122	Linear Algebra	2	36
	UCSMT123	Mathematics Practical based on UCSMT121 & UCSMT122	2	48

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

Semester	Course Code	Title of Course	No. of Credits	No. of Lectures
III	UCSMT231	Groups and Coding Theory	3	48
	UCSMT232	Numerical Techniques	3	48
	UCSMT233	Mathematics Practical Python Programming Language I	2	48
IV	UCSMT241	Computational Geometry	3	48
	UCSMT242	Operation Research	3	48
	UCSMT243	Mathematics Practical Python Programming Language II	2	48

Equivalence of the Old Syllabus with New Syllabus:

Semester	Old Course		New Course	
	F.Y.B.Sc.(Comp. Sci.)			
I	CSMT1101	Graph Theory	UCSMT111	Graph Theory
	CSMT1102	Algebra	UCSMT112	Matrix Algebra
	CSMT1103	Mathematics Practical based on CSMT1101 & CSMT1102	UCSMT113	Mathematics Practical based on UCSMT111 & UCSMT112
II	CSMT1201	Discrete Mathematics	UCSMT121	Discrete Mathematics
	CSMT1202	Calculus	UCSMT122	Linear Algebra
	CSMT1203	Mathematics Practical based on CSMT1201 & CSMT1202	UCSMT123	Mathematics Practical based on UCSMT121 & UCSMT122
	S.Y.B.Sc.(Comp. Sci.)			
III	CSMT2301	Linear Algebra	UCSMT231	Groups and Coding Theory
	CSMT2302	Numerical Analysis	UCSMT232	Numerical Techniques
	CSMT2303	Mathematics Practical I	UCSMT233	Mathematics Practical Python Programming Language I

Choice Based Credit System Syllabus (2022 Pattern)

Class:S.Y.B.Sc.(Computer Science). (Sem III)
Subject: Mathematics
Course: Groups and Coding Theory **Course Code:**UCSMT231

A) Course Objectives

1. To introduce concept of relation.
2. To introduce basic algebraic properties of groups.
3. Use algebraic techniques to construct efficient codes.
4. Apply Euclid's lemma to solve problems related to divisibility.
5. Provide a clear definition of groups and offer examples to illustrate the concept.
6. Introduce the fundamentals of public-key cryptography and its connection to group theory.
7. Enhance mathematical reasoning and logic through the application of abstract algebraic concepts.

B) Course Outcomes

1. Students will revise the concept of equivalence relations and learn to apply them to congruence relations on the set of integers.
2. Students will have a thorough understanding of integers, including the division algorithm, G.C.D. computation using the Euclidean algorithm, and the application of Euclid's lemma.
3. Students will understand the concept of groups, including binary operations, and will be able to identify and provide examples of groups.
4. Understand basic terminologies related to finite groups and subgroups, including the subgroup test and properties of cyclic groups.
5. Gain a thorough understanding of cosets, properties associated with them, and the Lagrange theorem that relates the order of a group to the order of its subgroups.
6. Apply group theory concepts to coding theory, including coding of binary information, error detection, decoding, error correction, and an introduction to public-key cryptography.
7. Develop problem-solving skills and critical thinking abilities through the application of mathematical concepts in coding theory and related areas.

TOPICS/CONTENT

Unit 1:Integers **[12 Lectures]**

1.1 Divisionalgorithm.
1.2 G.C.D. and Euclidean algorithm.
1.3 Euclid'slemma.
1.4 Equivalencerelation(revision), Congruencerelationonsetofintegers.
1.5 Equivalence class and partitions.

Unit 2: Groups **[08 Lectures]**

2.1 BinaryOperation
2.2 Group:DefinitionandExamples
2.3 ElementaryPropertiesofGroups

Unit 3: FiniteGroupsand Subgroups **[16 Lectures]**

3.1 Basic terminologies.
3.2 Subgroup test.
3.3 Cyclic groups.
3.4 Properties of cyclic groups.
3.5 Classification of subgroups of cyclic groups.
3.6 Permutation groups.
3.7 Properties of permutation groups.
3.8 Cosets.
3.9 Properties of cosets.
3.10 Lagrange theorem.

Unit 4: GroupsandCodingTheory **[12 Lectures]**

4.1 Coding ofBinaryInformationandErrordetection
4.2 DecodingandErrorCorrection
4.3 PublicKeyCryptography

Text Book:

1. J. A. Gallian, ContemporaryAbstractAlgebra, Narosa, 7th Edition
   Unit 1: Chapter 0
   Unit 2: Chapter 2
   Unit 3: Chapters 3, 4, 5 and 7

2. Bernard Kolman, Robert C. Busby and Sharon Ross, Discrete Mathematical Structures, Pearson Education Publication, 6th Edition.
   Unit 4: Chapter 11

Reference Book:

1. N. S. Gopalakrishnan, University Algebra, New Age International (P) Ltd, Publishers, 2nd Edition (1986).
2. P. B. Bhattacharya, S. K. Jain, S. R. Nagpaul, Basic Abstract Algebra, Cambridge University Press, 2nd Edition (1994).
3. I. N. Herstein, Topics in Algebra, Wiley, 2nd Edition.
4. J. H. van Lint, Introduction to Coding Theory, Springer.

Mapping of Program Outcomes with Course Outcomes

Class:S.Y.B.Sc.(Computer Science). (Sem III)
Subject: Mathematics
Course: Groups and Coding Theory **Course Code:**UCSMT231
Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

	Programme Outcomes (POs)						
Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1		2	3			2	
CO 2		3				3	3
CO 3			2			3	
CO 4		2				3	3
CO 5		1	2			2	
CO 6			3			2	2
CO 7		2	3			3	3

Justification for the mapping

PO2: Design / Development of solution

CO1:Enhancing mathematical proficiency, this course empowers students to master equivalence relations and adeptly apply them to congruence relations on the set of integers for robust solution design and development.

CO2: Students will gain a comprehensive grasp of integers, encompassing the division algorithm, Euclidean algorithm for G.C.D. computation, and the practical application of Euclid's lemma in solution design and development.
CO4: A solid grasp of finite group and subgroup terminologies, subgroup testing, and properties of cyclic groups is essential for designing and developing solutions with a strong foundation in abstract algebra and mathematical structures.
CO5: Understanding cosets and their properties, along with the Lagrange theorem, is essential for designing and developing solutions in group theory, providing a foundation to analyze and manipulate group structures efficiently.
CO7:Enhancing problem-solving skills and critical thinking by applying mathematical concepts in coding theory fosters effective design and development of innovative solutions.

PO3: Modern tool usage

CO1: Equipping students with a deep understanding of equivalence relations and their practical application to congruence relations on the set of integers through modern tool usage enhances their mathematical proficiency and problem-solving skills.
CO3:Enabling students to grasp the concept of groups and binary operations fosters a foundational understanding essential for recognizing and applying group theory in modern tool usage scenarios.

CO5:Mastering cosets and their associated properties, along with a profound grasp of the Lagrange theorem, enhances modern tool usage by providing a foundational understanding of group theory, facilitating efficient exploration and utilization of subgroup structures.
CO6:Applying group theory to coding theory enhances modern tool usage by providing a rigorous mathematical framework for efficient coding of binary information, error detection, decoding, error correction, and introducing foundational concepts essential for robust public-key cryptography.
CO7:Enhancing problem-solving skills and critical thinking by applying mathematical concepts in coding theory fosters proficiency in modern tool usage, essential for addressing complex challenges in diverse technological domains.

PO6: Individual and Team work

CO1: Enhancing mathematical proficiency, students will master equivalence relations, applying them adeptly to congruence relations on the integer set through a combination of individual and collaborative efforts.
CO2:Equipping students with a comprehensive grasp of integers, encompassing the division algorithm, G.C.D. computation through the Euclidean algorithm, and the practical application of Euclid's lemma, fosters adeptness in both individual problem-solving and collaborative teamwork.
CO3:Enhancing collaboration skills, students will grasp group theory concepts, identify binary operations, and exemplify group structures in both individual and team contexts, fostering a comprehensive understanding of mathematical and interpersonal dynamics.
CO4:Mastering basic finite group and subgroup terminologies, including the subgroup test and properties of cyclic groups, is essential for effective collaboration in both individual and team work within the realm of abstract algebra.
CO5:Mastering cosets and the associated properties, along with a profound grasp of the Lagrange theorem, enhances both individual and team proficiency in understanding group theory by establishing a crucial link between group and subgroup orders.
CO6:Applying group theory concepts to coding theory enhances both individual and team capabilities in designing efficient coding schemes for binary information, error detection, decoding, error correction, and introduces essential principles for public-key cryptography.
CO7:Enhancing problem-solving skills and critical thinking by applying mathematical concepts in coding theory fosters both individual and collaborative proficiency, essential for addressing complex challenges in diverse professional settings.

PO7: Innovation, employability and Entrepreneurial skills

CO2: Mastering integers and mathematical algorithms fosters critical thinking, problem-solving, and logical reasoning, essential for innovation, employability, and entrepreneurial success.

CO4: Understanding basic terminologies related to finite groups and subgroups, including the subgroup test and properties of cyclic groups, fosters mathematical literacy essential for innovation, employability, and entrepreneurial problem-solving in diverse fields.

CO6:Applying group theory to coding theory enhances innovation in information security by providing a mathematical framework for efficient coding, error detection, and correction, along with an introduction to public-key cryptography, fostering employability and entrepreneurial skills in the rapidly evolving digital landscape.

CO7:Enhancing problem-solving and critical thinking through the practical application of mathematical concepts in coding theory cultivates essential skills vital for innovation, employability, and entrepreneurial success.

Choice Based Credit System Syllabus (2022 Pattern)

Mapping of Program Outcomes with Course Outcomes

Class:S.Y.B.Sc.(Computer Science). (Sem III)
Subject: Mathematics
Course: Numerical Techniques **Course Code:** UCSMT232
Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

A) Course Objectives

1. To introduce calculus of finite difference.
2. To introduce methods in numerical integration.
3. To solve ordinary differential equations using numerical techniques.
4. Apply the Trapezoidal Rule, Simpson's one-Third Rule, and Simpson's Three-Eight Rule for numerical integration with emphasis on accuracy and efficiency.
5. Apply Euler's Method, Euler's Modified Method, and Runge-Kutta Methods for solving ordinary differential equations numerically.
6. Identify conditions under which numerical methods provide reliable solutions.
7. Analyze and interpret the convergence properties and stability of numerical methods employed in solving mathematical problems.

B) Course Outcomes

1. Understand the significance of numerical accuracy and precision in computations.
2. Apply the False Position Method to solve algebraic equations and understand its convergence behavior.
3. Understand and apply forward, backward, and central differences in numerical calculations.
4. Apply Central Difference Formulae, including Gauss Forward and Backward Difference Formulas, and Bessel's Interpolation Formula.
5. Apply Simpson's One-Third Rule and Simpson's Three-Eight Rule for accurate numerical integration.
6. Apply divided differences and Newton's Divided Difference Formula for interpolating values.
7. Apply the Runge-Kutta Method for more accurate and efficient numerical solutions to ordinary differential equations.

TOPICS/CONTENT

Unit 1: AlgebraicandTranscendentalEquation [08 Lectures]

1.1 IntroductiontoErrors
1.2 FalsePositionMethod
1.3 Newton-RaphsonMethod

Unit 2:CalculusofFiniteDifferencesandInterpolation [16 Lectures]

2.1 Differences
2.2 ForwardDifferences
2.3 BackwardDifferences
2.4 CentralDifferences
2.5 OtherDifferences(δ,μoperators)
2.6 PropertiesofOperators
2.7 Relation betweenOperators
2.8 Newton'sGregoryFormulaforForwardInterpolation
2.9 Newton'sGregoryFormula forBackwardInterpolation
2.10 Lagrange'sInterpolationFormula
2.11 DividedDifference
2.12 Newton'sDividedDifferenceFormula

Unit 3: NumericalIntegration [12 Lectures]

3.1 GeneralQuadratureFormula
3.2 TrapezoidalRule
3.3 Simpson'sone-ThirdRule
3.4 Simpson'sThree-Eight Rule

Unit 4:NumericalSolutionofOrdinary DifferentialEquation [12 Lectures]

4.1 Euler'sMethod
4.2 Euler'sModifiedMethod
4.3 Runge-KuttaMethods

Text Book:

1. A.K.JaiswalandAnjuKhandelwal, AtextbookofComputerBased Numericaland StatisticalTechniques, New AgeInternationalPublishers.

Unit1: Chapter2:Sec.2.1,2.5,2.7
Unit 2: Chapter3:Sec.3.1,3.2,3.4,3.5,Chapter4: Sec.4.1,4.2,4.3, Chapter 5:Sec.5.1,5.2,5.4,5.5
Unit 3: Chapter6:Sec.6.1,6.3,6.4,6.5,6.6,6.7
Unit4: Chapter7:Sec.7.1,7.4,7.5,7.6

Reference Book:

1. S.S.Sastry;IntroductoryMethodsofNumericalAnalysis,3rdedition,Prentice HallofIndia,1999.
2. H.C.Saxena;FinitedifferencesandNumericalAnalysis,S.ChandandCompany.
3. K.E.Atkinson;AnIntroductiontoNumericalAnalysis, WileyPublications.
4. Balgurusamy;NumericalAnalysis.

Mapping of Program Outcomes with Course Outcomes

Class:S.Y.B.Sc.(Computer Science). (Sem III)
Subject: Mathematics
Course: Numerical Techniques **Course Code:** UCSMT232
Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

	Programme Outcomes (POs)						
Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1		2	3			3	
CO 2			2			2	3
CO 3	3		1			3	
CO 4			1			2	3
CO 5			3			3	2
CO 6			3			2	2
CO 7	2		1			3	

Justification for the mapping

PO1: Computer Knowledge
CO3: Student understanding and applying forward, backward, and central differences in numerical calculations acquire essential skills in computer knowledge, enabling them to develop robust algorithms, optimize computational efficiency, and solve real-world problems with precision.
CO7: Student applying the Runge-Kutta Method for more accurate and efficient numerical solutions to ordinary differential equations in computer knowledge acquire a critical skill set, enhancing their ability to model dynamic systems, simulate real-world scenarios, and optimize computational accuracy in various applications.

PO2: Design / Development of solution

CO1: Student understanding the significance of numerical accuracy and precision in computations during the design/development of solutions gain a foundational awareness critical for ensuring reliable and robust algorithms, minimizing errors, and optimizing the performance of computational solutions.

PO3: Modern tool usage
CO1: Student understanding the significance of numerical accuracy and precision in computations in modern tool uses develop a critical foundation for optimizing algorithms, minimizing errors, and ensuring the reliability of computational solutions across diverse applications.
CO2: Student applying the False Position Method to solve algebraic equations and understanding its convergence behavior in modern tool uses gain a valuable skill set, enabling efficient and reliable numerical solutions with insights into convergence characteristics for enhanced problem-solving.

CO3: Student understanding and applying forward, backward, and central differences in numerical calculations in modern tool uses acquire foundational skills essential for developing accurate algorithms, optimizing computational efficiency, and solving real-world problems with precision.
CO4: Student applying Central Difference Formulae, including Gauss Forward and Backward Difference Formulas, and Bessel's Interpolation Formula in modern tool uses, develop advanced numerical skills, facilitating precise data interpolation, algorithm optimization, and efficient problem-solving in diverse computational applications.
CO5: Student applying Simpson's One-Third Rule and Simpson's Three-Eight Rule for accurate numerical integration in modern tool uses develop advanced skills, ensuring precise calculation of definite integrals and enhancing computational accuracy in diverse applications.
CO6: Student applying divided differences and Newton's Divided Difference Formula for interpolating values in modern tool uses acquire crucial skills for accurate data interpolation, facilitating effective modeling and analysis in diverse computational applications.
CO7: Student applying the Runge-Kutta Method for more accurate and efficient numerical solutions to ordinary differential equations in modern tool uses acquire advanced computational skills, enabling precise modeling and simulation of dynamic systems across diverse applications.

PO6: Individual and Team work

CO1:Student understanding the significance of numerical accuracy and precision in individual and team work cultivate essential skills for collaborative problem-solving, effective communication, and reliable decision-making, ensuring the success of diverse projects and tasks.

CO2:Student applying the False Position Method to solve algebraic equations and understanding its convergence behavior in individual and team work acquire critical numerical skills, facilitating collaborative problem-solving and informed decision-making across a range of applications and projects.

CO3:Student understanding and applying forward, backward, and central differences in numerical calculations in individual and team work develop foundational skills, promoting effective collaboration, problem-solving, and optimization of computational methods across diverse projects and tasks.

CO4:Student applying Central Difference Formulae, including Gauss Forward and Backward Difference Formulas, and Bessel's Interpolation Formula in individual and team work, acquire advanced numerical skills, fostering collaborative problem-solving and efficient data interpolation across a variety of projects and tasks.

CO5:Student applying Simpson's One-Third Rule and Simpson's Three-Eight Rule for accurate numerical integration in individual and team work develop advanced skills, promoting collaborative problem-solving and precise calculations in diverse projects and tasks that involve numerical analysis.

CO6:Student applying divided differences and Newton's Divided Difference Formula for interpolating values in individual and team work acquire essential skills, fostering collaborative problem-solving and efficient data interpolation across various projects and tasks in diverse domains.

CO7:Students applying the Runge-Kutta Method for more accurate and efficient numerical solutions to ordinary differential equations in individual and team work gain advanced computational skills, enhancing collaborative problem-solving and enabling precise modeling of dynamic systems across diverse projects and tasks.

PO7: Innovation, employability and Entrepreneurial skills

CO2: Student applying the False Position Method to solve algebraic equations and understanding its convergence behavior foster innovation, employability, and entrepreneurial skills by gaining valuable numerical problem-solving abilities essential for addressing real-world challenges and optimizing solutions in various professional contexts.

CO4: Student applying Central Difference Formulae, including Gauss Forward and Backward Difference Formulas, and Bessel's Interpolation Formula in innovation, employability, and entrepreneurial skills develop advanced computational capabilities, enabling them to tackle complex problems, optimize algorithms, and contribute to innovative solutions in diverse professional settings.

CO5: Student applying Simpson's One-Third Rule and Simpson's Three-Eight Rule for accurate numerical integration in innovation, employability, and entrepreneurial skills develop advanced computational expertise, enhancing their ability to optimize algorithms, make informed decisions, and contribute to innovative solutions in various professional contexts.

CO6: Student applying divided differences and Newton's Divided Difference Formula for interpolating values in innovation, employability, and entrepreneurial skills acquire crucial numerical abilities, enabling them to contribute to innovative solutions, optimize algorithms, and address real-world challenges in various professional contexts.

Choice Based Credit System Syllabus (2022 Pattern)

Mapping of Program Outcomes with Course Outcomes

Class:S.Y.B.Sc.(Computer Science). (Sem III)
Subject: Mathematics
Course: Mathematics Practical Python Programming Language I **Course Code:** UCSMT233

A) Course Objectives

1. Define variables using assignment statements and comprehend the types: int, float, and str.
2. Apply arithmetic operators (+, -, /, *, **, %) and understand their precedence using PEMDAS rules.
3. Utilize mathematical functions from the math and cmath modules.
4. Define tuples, access elements using the index and slice operators, perform tuple assignments, and use tuples as return values.
5. Perform matrix operations including addition, subtraction, multiplication, and matrix powers and inverses.
6. Apply Python programming to solve problems related to linear algebra, numerical methods, and mathematical computations.
7. Develop problem-solving skills through hands-on coding exercises.

B) Course Outcomes

1. Students will be able to install Python and understand the basic values and types such as int, float, and str.
2. Understand and apply various operators, operands, and the rules of precedence (PEMDAS) in Python.
3. Apply comparison operators (==, !=, >, <, >=, <=) and logical operators (and, or, not) in Boolean expressions.
4. Develop a strong understanding of functions, including user-defined functions, parameters, and arguments.
5. Apply numerical methods such as Newton-Raphson Method and False Position (RegulaFalsi) Method for root finding.
6. Apply numerical methods in solving real-world problems and mathematical challenges.
7. Develop programming skills to implement numerical algorithms and analyze their efficiency.

TOPICS/CONTENT

Unit1:IntroductiontoPython

1.1 InstallationofPython
1.2 Valuesandtypes:int,floatandstr,
1.3 Variables:assignment statements,printingvariablevalues,typesofvariables.
1.4 Operators,operandsandprecedence:+,-,/,*,**, %PEMDAS(Rulesofprecedence)
1.5 Stringoperations:+:Concatenation,*:Repetition
1.6 Booleanoperator:
    1.6.1 Comparisonoperators: ==,!=,>,=,<=
    1.6.2 Logicaloperators:and,or,not
1.7 Mathematicalfunctionsfrommath,cmathmodules.
1.8 Keyboardinput:input()statement

Unit 2:String,list,tuple

2.1 Strings:
    2.1.1 Length(Lenfunction)
    2.1.2 Stringtraversal:Usingwhilestatement,Usingforstatement
    2.1.3 Stringslice
    2.1.4 Comparisonoperators(>,<, ==)
2.2 Lists:
    2.2.1 Listoperations
    2.2.2 Useofrangefunction
    2.2.3 Accessinglistelements
    2.2.4 Listmembershipandforloop
    2.2.5 Listoperations
    2.2.6 Updatinglist:addition, removalorupdatingofelementsofalist
2.3 Tuples:
    2.3.1 Definingatuple,
    2.3.2 Indexoperator,
    2.3.3 Sliceoperator,
    2.3.4 Tupleassignment,
    2.3.5 Tupleasareturnvalue

Unit 3:Iterationsand Conditionalstatements

3.1 Conditionalandalternativestatements,ChainedandNestedConditionals:if,if-else,if-elif-else,nestedif,nestedif-else
3.2 Loopingstatementssuchaswhile,for etc,Tablesusingwhile.
3.3 Functions:
    3.3.1 Callingfunctions:type,id
    3.3.2 Typeconversion:int,float,str
    3.3.3 Compositionoffunctions
    3.3.4 Userdefined functions,Parametersandarguments

Unit 4:LinearAlgebra

4.1 Matrixconstruct,eye(n),zeros(n,m)matrices
4.2 Addition,Subtraction,Multiplicationofmatrices,powersand inversofa matrix.
4.3 AccessingRowsandColumns,DeletingandInsertingRowsandColumns
4.4 Determinant,reducedrowechelonform,nullspace,columnspace,Rank
4.5 Solvingsystemsoflinearequations(GaussEliminationMethod, GaussJordanMethod,LU-decompositionMethod)

4.6 Eigenvalues,Eigenvectors,andDiagonalization

Unit 5:NumericalmethodsinPython

5.1 RootsofEquations

5.2 Newton-RaphsonMethod

5.3 FalsePosition(RegulaFalsi)Method

5.4 NumericalIntegration:

5.1.1 TrapezoidalRule,

5.1.2 Simpson's1/3rdRule,

5.1.3 Simpson's3/8thRule

Text Books:-

1. Downey,A.et al.,Howtothinklikea ComputerScientist: Learning withPython,JohnWiley,2015.
   Sections:1,2,3
2. RobertJohansson, IntroductiontoScientificComputinginPython
   Section:4

ReferenceBooks:-

1. LambertK.A.,FundamentalsofPython-FirstPrograms,CengageLearningIndia,2015.
2. Guzdial,M.J., IntroductiontoComputingandProgramminginPython, Pearson India.
3. Perkovic,L.,IntroductiontoComputingUsingPython, 2/e, JohnWiley,2015.
4. Zelle,J.,PythonProgramming:AnIntroductiontoComputerScience,Franklin,Beedle &Associates Inc.
5. SandroTosi,MatplotlibforPythonDevelopers,PacktPublishingLtd.(2009)

Mapping of Program Outcomes with Course Outcomes

Class:S.Y.B.Sc.(Computer Science). (Sem III)
Subject: Mathematics
Course: Mathematics Practical Python Programming Language I **Course Code:** UCSMT233
Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

	Programme Outcomes (POs)						
Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1			3				
CO 2	3					2	2
CO 3	3	2				1	
CO 4	3		1				3
CO 5						3	
CO 6							3
CO 7	1		3			2	

Justification for the mapping

PO1: Computer Knowledge
CO2: Understanding and applying operators, operands, and the rules of precedence (PEMDAS) in Python is crucial for accurate and efficient computation, facilitating precise execution of mathematical expressions and logical operations in computer programming.
CO3: Comparison and logical operators in Boolean expressions enable precise evaluation of conditions, facilitating effective decision-making and control flow in computer programs.
CO4: Developing a strong understanding of functions, including user-defined functions, parameters, and arguments in Computer Knowledge, enhances problem-solving and modular programming skills, facilitating efficient code organization and reusability.
CO7: Enhancing programming skills for numerical algorithms enables efficient implementation and analysis, fostering proficiency in computational problem-solving within the realm of computer knowledge.

PO2: Design / Development of solution

CO3:Comparison and logical operators enable precise control and decision-making in Boolean expressions, enhancing the design and development of solutions by facilitating conditional evaluations and logical flow within the program.

PO3: Modern tool usage
CO1: Enabling students to install Python and grasp fundamental data types cultivates essential skills for modern tool usage, fostering a foundational understanding of programming values and types.
CO4: Developing a strong understanding of functions in modern tool usage is essential for efficient and modular programming, enabling the creation of reusable code through user-defined functions with clear parameter and argument definitions.
CO7:Developing programming skills for numerical algorithms allows for efficient implementation and analysis, enabling effective utilization of modern tools for solving complex mathematical problems in various fields.

PO6: Individual and Team work

CO2: Understanding and applying operators, operands, and the rules of precedence (PEMDAS) in Python is essential for accurate and efficient mathematical and logical operations, promoting precision and collaboration in both individual and team programming endeavors.
CO3:Comparison and logical operators enhance Individual and Team work by facilitating precise evaluation of conditions, fostering effective decision-making, and promoting seamless collaboration in programming tasks.
CO4:Applying numerical methods like Newton-Raphson and False Position enhances precision and efficiency in root finding, facilitating both individual and team efforts in solving complex mathematical problems with iterative accuracy.
CO7:Enhancing programming skills for numerical algorithm implementation fosters adeptness in problem-solving, enabling efficient analysis and collaboration in both individual and team settings within diverse computational contexts.

PO7: Innovation, employability and Entrepreneurial skills

CO2: Understanding and applying various operators, operands, and the rules of precedence (PEMDAS) in Python enhances innovation by facilitating efficient algorithmic problem-solving, boosts employability through foundational programming skills, and fosters entrepreneurial skills by enabling the development of customized solutions and automation.

CO4: Proficiency in functions, encompassing user-defined functions, parameters, and arguments, fosters innovation by empowering individuals to create modular, scalable solutions, enhances employability through versatile programming skills, and fuels entrepreneurial success by enabling efficient problem-solving and product development.

CO6:Numerical methods enable precise and efficient solutions to complex real-world problems, fostering innovation, enhancing employability, and empowering entrepreneurial endeavors through rigorous mathematical analysis and computation.