

Program structure & Detailed Syllabus

For

Under Graduate Programme (B.Tech)

ARTIFICIAL INTELLIGENCE & DATA SCIENCE
(Applicable For Batches Admitted From 2020 – 2021)



VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)

DUVVADA - VISAKHAPATNAM – 530 049

(An Autonomous Institute, Accredited by NAAC, Affiliated to JNTUK,
Kakinada, AP)

S. No.	Course Code	Course Title	L	T	P	C
1	1000202103	Statistics for Data Science-II	3	1	0	3
2	1054202100	Data Engineering	3	0	0	3
3	1005202104	Programming Essentials in Python	3	0	0	3
4	1005202103	Software Engineering	3	0	0	3
5	1000202100	Discrete Mathematical Structures	3	1	0	3
6	1054202110	Data Engineering Lab	0	0	3	1.5
7	1005202113	Programming Essentials in Python Lab	0	0	3	1.5
8	1005202112	Unified Modeling Language Lab	0	0	3	1.5
9	1054202180	Competitive Programming	0	0	4	2
10	1000202120	Life Skills	2	0	0	0
Total Credits						21.5

Detailed Syllabus

II Year- I Semester

Program Structure and Detailed Syllabus (VR 20)

II Year – I Semester	STATISTICS FOR DATA SCIENCE-II	L	T	P	Credits
100202103		3	1	0	3

Course Objectives:

- ✓ To understand basic theoretical knowledge about fundamental principles for statistical inference.
- ✓ To understand different statistical test applied to samples to infer conclusions.

✓ **COURSE OUTCOMES:**

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Conduct hypothetical testing for large samples and small samples	1	1
		2	2
		3	3
CO2	Calculate, and interpret, the correlation coefficient and regression models	1	2
		2	3
CO3	Interpret production or service quality by using different quality control charts.	3	2
		5	3
CO4	Analyze the time series data using different techniques	1	1
		2	2
		3	3
		4	3
		12	3

Unit-I

Test of Hypothesis-I:

[8 Hours]

Concept and definition of statistical hypothesis, Type I and Type II errors. One tail, two-tail tests, level of significance, P –Values in Decision Making, confidence intervals.

Large sample tests: Hypothesis testing of means, proportions and difference between means and proportions.

Unit-II

Test of Hypothesis-II:

[10 Hours]

Small Sample tests: Students' t-distribution: single mean, difference of means, paired t-test for difference of means, F-distribution and $X^2 X^2$ test. Test of independence of attributes - ANOVA for one-way and two-way classified data.

Unit-III

Correlation and Regression: [10 Hours]

The method of least squares – Inferences based on the least square's estimations, Regression: definition, linear regression, multiple regression and curve linear regression, Correlation: definition, correlation coefficient, rank correlation and correlation for bivariate distributions

Unit-IV

Statistical Quality Control methods: [9 Hours]

Introduction, types of control charts: control charts for variables and attributes, Methods for preparing control charts and distribution curve.

Unit-V

Statistical Methods in Time series Analysis data: [10 Hours]

Introduction, importance of Time series analysis, Methods in time series data analysis

Text Books:

- 1. Fundamentals of Mathematical Statistics by S.C. Gupta and V.K. Kapoor
- 2. Probability and Mathematical statistics by Prasanna Sahoo
- 3. Introduction to Time Series and Forecasting, Second Edition, by Peter J. Brockwel
Richard A. Davis

Reference Books:

- 1) Yanchang Zhao, "R and Data Mining: Examples and Case Studies", Elsevier, 1st Edition, 2012

II Year – I Semester		L	T	P	C
1054202100	DATA ENGINEERING	3	0	0	3

COURSE OBJECTIVES:

- To Identify the scope and essentiality of Data.
- To Analyse data, choose relevant models and algorithms for respective applications.
- To Provide students with theoretical knowledge and practical skills in the use of database and database management systems in information technology applications
- To Develop research interest towards advances in data science.

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Understand Data Science fundamentals, Data Representation.	PO1, PO2, PO12	3
CO2	Create, maintain and manipulate a relational database using SQL.	PO1, PO2, PO12	3
CO3	Design and build database system for a given real world problem.	PO1, PO2, PO3, PO12	3
CO4	Identify appropriate data Processing techniques to solve real world problems.	PO1, PO2, PO3, PO12	3

**Strength of mapping (Intensity Scale) – 1(Lightly mapped), 2(Moderately mapped), 3(Heavily mapped)

UNIT- I

INTRODUCTION TO DATA SCIENCE

Introduction to Data Science – Evolution of Data Science – Data Engineering Ecosystem, Data Engineering Lifecycle -Data Science Roles – Data Engineering in Action-Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues-Data Sources, Data Storages- Traditional File System. **[6 HOURS]**

UNIT-II

DATA MODELS AND REPRESENTATION

Data model, structured data, unstructured data, semi-structured data and Data representation, feature representations, encoded representations, Spatial data representation, Time-series data representations. **[6 HOURS]**

UNIT-III

ACCESSING DATA- RELATIONAL DATABASE

E-R Model: Overview of Database Design, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model Introduction to the Relational Model, Relational model constraints over relations. Relational Algebra and calculus.

[8 HOURS]

UNIT-IV

ACCESSING DATA- SCHEMA REFINEMENT

SQL Queries: The Form of Basic SQL Query, Union, Intersect and Except-Nested Queries- Aggregative Operators- Group by and Having Clauses-Null Values-Outer Joins.

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form (BCNF), Lossless join and dependency preserving decomposition, Fourth normal form(4NF), De-normalization.

[8 HOURS]

UNIT-V

DATA ANALYSIS TOOLS

Charts: Bar Chart/Graph. Pie Chart. Line Graph or Chart, Histogram Chart, Area Chart, Check sheet, Control chart, Gantt chart, Plots: Dot Graph or Plot, Scatter Plot, Box and whisker plot, Design of experiments (DOE).

[10 HOURS]

Text Books:

1. The Data Engineering Cookbook by Andreas Kretz
2. Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA.
3. R for data science : Import, Tidy, Transform, Visualize, And Model Data Paperback – 20 January 2017 by Hadley Wickham, Garrett Grolemund

Reference Books:

1. Introduction to Database Systems, 8/e C J Date, PEA
2. The Database book principles & practice using Oracle/MySQL Narain Gehani, University Press.
3. Data Engineering with Python: Work with Massive Datasets to Design Data Models and Automate Data Pipelines Using Python

Introduction to Database Systems, 8/e C J Date, PEA

E-Books: <https://github.com/andkret/Cookbook>

NPTEL/MOOC:

1. <https://www.coursera.org/learn/introduction-to-data-engineering#syllabus>
2. <https://www.udacity.com/course/data-engineer-nanodegree--nd027>
3. <https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs09/>

II Year – I Semester		L	T	P	C
1005202104	Programming Essentials in Python	3	0	0	3

COURSE OBJECTIVES:

1. To learn about Python programming language syntax, semantics, and the runtime environment.
2. To be familiarized with universal computer programming concepts like data types, containers.
3. To be familiarized with general computer programming concepts like conditional execution, loops & functions.
4. To be familiarized with general coding techniques and object-oriented programming

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Develop essential programming skills in computer programming concepts like data types, containers	PO1	3
CO2	Apply the basics of programming in the Python language	PO1 PO2	3 3
CO3	Solve coding tasks related to the fundamental notions and techniques used in object-oriented programming.	PO5	2 3
CO4	Solve coding tasks related OOPS, and Multithreading	PO12	3

UNIT- I

INTRODUCTION TO PYTHON:

History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation. Types, Operators and Expressions: Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations. Control Flow-: if, if-elif-else, for, while, break, continue, pass.

[6 Hours]

UNIT- II

STRINGS and DATA STRUCTURES:

Strings: Strings and text files, String manipulations: subscript operator, indexing, slicing a string; strings and number system: converting strings to numbers and vice versa. Binary,

Program Structure and Detailed Syllabus (VR 20)

octal, hexadecimal numbers, Data Structures: Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries.

[8 Hours]

UNIT- III

FUNCTIONS:

Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables. Modules: Creating modules, import statement, from, name spacing, Python packages Introduction to PIP, Installing Packages via PIP, Using Python Packages. [8 Hours]

UNIT- IV

INTRODUCTION TO OOPS:

Classes and Objects: Introduction, classes and objects, class method and self-argument, init() method, class and object variables, del() method, other special methods, public and private data members, private methods, calling a class method from another class method, built-in class attributes, garbage collection, class and static methods, Inheritance: Introduction, inheriting classes in python, types of inheritance, composition/containership/complex objects, abstract classes and interfaces, Meta class. [8 Hours]

UNIT- V

OPERATOR OVERLOADING AND EXCEPTION HANDLING

Operator Overloading: Introduction, implementing operator overloading, reverse adding, overriding __getitem__() and __setitem__() methods, overriding the in operator, overriding miscellaneous functions, overriding the _call__() method.

Error and Exception Handling: Introduction to errors and exceptions, handling exceptions, multiple except blocks, multiple exceptions in a single block, except block without exception, the else clause, raising exceptions, built-in and user-defined exceptions, the finally block.

[10 Hours]

Text Books:

1. Reema Thareja, "Python Programming Using Problem Solving Approach", Oxford University Press, 2019.
2. Zed Shah, "Learn Python The Hard Way", Third edition, Addison-Wesley, 2013.

Reference Books:

1. Charles Severance, "Python for Informatics- Exploring Information", 1st edition Shroff Publishers, 2017.
2. John V. Guttag, "Introduction to Computation and Programming Using Python", The MIT Press,
3. W.Chun, "Core Python Programming", 2nd Edition, Prentice Hall, 2006.
4. Core Python Programming - Covers Fundamentals to Advanced Topics Like OOPS, Exceptions, Data Structures, Files, Threads, Networking, GUI, DB Connectivity and

Data Science Second Edition (English, Paperback, Rao R. Nageswara)

E-Books: <https://www.python.org/doc/>

NPTEL/MOOC:

1. <https://nptel.ac.in/courses/106/106/106106182/>
2. <https://nptel.ac.in/courses/106/106/106106145/>

II Year – I Semester		L	T	P	C
1005202103	Software Engineering	3	0	0	3

COURSE OBJECTIVES:

1. To understand the software life cycle models.
2. To understand the software requirements and SRS document.
3. To understand the importance of modeling and modeling languages.
4. To design and develop correct and robust software products.
5. To understand the quality control and how to ensure good quality software.
6. To understand the planning and estimation of software projects.
7. To understand the implementation issues, validation and verification procedures.
8. To understand the maintenance of software

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Apply the appropriate process models for the application development of SDLC	PO1 PO2 PO3 PO5 PO8	1 2 2 1 2
CO2	Understand the phases of SDLC from requirement gathering phase to design phase via Analysis Phase	PO1 PO2	2 1
CO3	Analyzing the strategies for coding and testing phase in Software product development	PO1 PO2 PO3	2 2 2
CO4	Apply the knowledge about estimation and maintenance of software systems and modeling the software project by using CASE tools	PO1 PO2 PO3 PO5	2 2 1 1

UNIT- I

Software and Software Engineering: The Nature of Software, The Unique Nature of Web Apps, Software Engineering, Software Process, Software Engineering Practice, Software Myths, Professional and ethical responsibility.

Process Models and Agile Development: Generic Process Models like Waterfall Models, Evolutionary Process Model, V-Model, Agile Model, etc. Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Terminology, Product and Process. Agile process-Extreme programming Process. **[10 Hours]**

UNIT-II

Program Structure and Detailed Syllabus (VR 20)

Requirements Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specification (SRS), Formal System Specification.

Software Design: Overview of the Design Process, How to Characterize of a Design? Cohesion and Coupling, Layered Arrangement of Modules, Approaches to Software Design. [8 Hours]

UNIT-III

Function-Oriented Software Design: Overview of SA/SD Methodology, Structured Analysis, Developing the DFD Model of a System, Structured Design, Detailed Design, Design Review, over view of Object Oriented design.

User Interface Design: Characteristics of Good User Interface, Basic Concepts, Types of User Interfaces, Fundamentals of Component-based GUI Development, A User Interface Design Methodology.

Coding And Testing: Coding, Code Review, Software Documentation, Testing and Test Case, Unit Testing, Black-Box Testing, White-Box Testing, Debugging, Program Analysis Tool, Integration Testing, Testing Object-Oriented Programs, System Testing, Some General Issues Associated with Testing. [10 Hours]

UNIT-IV

Software Reliability And Quality Management: Software Reliability, Statistical Testing, Software Quality, Software Quality Management System, ISO 9000, SEI Capability Maturity Model.

Computer Aided Software Engineering: Case and its Scope, Case Environment, Case Support in Software Life Cycle, Other Characteristics of Case Tools, Towards Second Generation CASE Tool, Architecture of a Case Environment [9 Hours]

UNIT-V

Software Maintenance: Software maintenance, Maintenance Process Models, Maintenance Cost, Software Configuration Management, Devops.

Software Reuse: what can be reused? Why almost No Reuse So Far? Basic Issues in Reuse Approach, Reuse at Organization Level. [8 Hours]

Text Books:

1. Software engineering A practitioner's Approach, Roger S. Pressman, Seventh Edition Mc Graw Hill International Edition.
2. Fundamentals of Software Engineering, Rajib Mall, Third Edition, PHI.
3. Software Engineering, Ian Sommerville, Ninth edition, Pearson education

Reference Books:

1. Software Engineering: A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
2. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
3. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

II Year – I Semester		L	T	P	C
1000202100	DISCRETE MATHEMATICAL STRUCTURES (Common for CSE, IT and AI&DS)	3	1	0	3

COURSE OBJECTIVES:

- To introduce the algorithmic approach to the solution of problems, which is fundamental in discrete mathematics and this approach reinforces the close ties between this discipline and the area of computer science.
- To introduce basic logical connectives and inference theory.
- To Familiarise closed form solution of linear recurrence relations by various methods.
- To perform the operations associated with sets, functions and relations.
- To Bring awareness of basic concepts of graphs and explaining related algorithms.

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Develop reasoning skills using Mathematical Logic concepts.	1	2
		2	2
		3	3
CO2	Evaluate the solutions for various problems using recurrence relations	1	1
		2	2
CO3	Construct Hasse diagrams and Understand the concept of Algebraic Structures	1	1
		2	2
		3	2
CO4	Apply the concepts of graph theory for a given problem.	1	1
		2	2
		3	3
		12	3

UNIT- I

Mathematical Logic

[12 Hours]

Propositional Logic: Connectives- negation, conjunction, disjunction, conditional and bi-conditional, well-formed formulae, tautologies, equivalence of formulae, tautological implications, Disjunctive and Conjunctive normal forms, Rules of inference and examples, Consistency of premises.

Predicative Logic.; Statement Functions, Variables and Quantifiers, Free and Bounded variables, Inference theory for predicative logic.

UNIT- II

Recurrence Relations

[8 Hours]

Recurrence relations: Recurrence relations, solving homogeneous linear recurrence relations by characteristic roots method, solving non homogeneous linear recurrence relations.

UNIT- III

Sets, Relations and Algebraic Structures

[12 Hours]

Sets: Sets, Operations on Sets, Principles of Inclusion–Exclusion, Pigeonhole Principle and its Application

Relations: Definition, representation, types of relations: equivalence relation, equivalence class, partial order, Hasse Diagram and total order relations.

Functions: Definition, types of functions: surjective , injective and bijective .

Algebraic Structures: Binary operations, Algebraic structures, Group, Abelian Group, Subgroups, Lagrange's theorem on finite groups.

UNIT- IV

Unit IV: Graph Theory

[10 Hours]

Graph theory: Definitions, finite and infinite graphs, incidence and degree, isolated and pendant vertices, isomorphism, sub graphs, connected and disconnected graphs, simple graph, complete graph, bipartite graph, complete bipartite graph, planar graph, Isomorphic Graphs, Euler formula(without proof) and Graph colouring, Walk, path and circuit, Euler graph, Hamiltonian Graph.

UNIT- V

Unit V : Trees

[10 Hours]

Trees: Some properties of trees, rooted and binary trees, spanning trees, BFS & DFS Algorithms, Minimal spanning trees, Kruskal's algorithm

Text Books:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill, 1997.
2. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, K. H. Rosen, 7th Edition, Tata McGraw Hill.
3. Joe L. Mott, Abraham Kandel and T. P. Baker, Discrete Mathematics for computer scientists & Mathematicians, 2/e, Prentice Hall of India Ltd, 2012.

Reference Books:

1. S. Santha and E. V. Prasad Mathematical Foundation For Computer Science , Cengage,2017.
2. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby, Sharon Cutler Ross, PHI.
3. Discrete Mathematics, S. K. Chakraborty and B.K. Sarkar, Oxford, 2011.
4. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rdEdition, Tata McGraw

Program Structure and Detailed Syllabus (VR 20)

II Year – I Semester		L	T	P	C
1054202110	DATA ENGINEERING LAB	0	0	3	1.5

COURSE OBJECTIVES:

- To Identify the scope and essentiality of Data.
- To Analyse data, choose relevant models and algorithms for respective applications.
- To Provide students with theoretical knowledge and practical skills in the use of database and database management systems in information technology applications
- To Develop research interest towards advances in data science.

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Understand Data Science fundamentals, Data Representation.	PO1, PO2, PO12	3
CO2	Create, maintain and manipulate a relational database using SQL.	PO1, PO2, PO12	3
CO3	Design and build database system for a given real world problem.	PO1, PO2, PO3, PO12	3 3
CO4	Identify appropriate data mining algorithms to solve real world problems	PO1, PO2, PO3, PO12	3

**Strength of mapping (Intensity Scale) – 1(Lightly mapped), 2(Moderately mapped), 3(Heavily mapped)

LIST OF EXPERIMENTS

S.No.	Name of the experiment	Skill
1.	Exercise – 1 Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.	Introduction
2.	Exercise – 2 Queries (along with sub-Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example: - Select the roll number and name of the student who secured fourth rank in the class.	Queries
3.	Exercise – 3 Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.	

Program Structure and Detailed Syllabus (VR 20)

4.	Exercises –4 Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc,round, to_char, to_date).	Queries
5.	Exercise -5 Write an SQL query to implement JOINS.	Joins
6.	Exercise -6 Data Visualization using Boxplot Plotting Framework using R. Data Visualization using Histogram Plotting Framework using R.	Data Visualization
7.	Exercise -7 Data Visualization using Line Graph Plotting Framework using R. Data Visualization using Scatterplot Plotting Framework using R.	Data Visualization
8.	Exercise -8 Data Visualization using Pie Chart Plotting Framework using R. Data Visualization using Bar Chart Plotting Framework using R.	Data Visualization
9.	Exercise -9 Application to adjust the Number of Bins in the Histogram using R Language/ Python Programming.	Data Visualization
10.	Exercise -10 Application to analyse Stock Market Data using R Language/ Python Programming.	Data Visualization

Text Books:

1. The Data Engineering Cookbook by Andreas Kretz
2. Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA.
3. R for data science : Import, Tidy, Transform, Visualize, And Model Data Paperback – 20 January 2017 by Hadley Wickham, Garrett Golemund

Reference Books:

1. Introduction to Database Systems, 8/e C J Date, PEA
2. The Database book principles & practice using Oracle/MySQL Narain Gehani, University Press.
3. Data Engineering with Python: Work with Massive Datasets to Design Data Models and Automate Data Pipelines Using Python.

Introduction to Database Systems, 8/e C J Date, PEA

E-Books: <https://github.com/andkret/Cookbook>

NPTEL/MOOC:

1. <https://www.coursera.org/learn/introduction-to-data-engineering#syllabus>
2. <https://www.udacity.com/course/data-engineer-nanodegree--nd027>
3. <https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs09/>

Program Structure and Detailed Syllabus (VR 20)

II Year – I Semester		L	T	P	C
1005202113	Programming Essentials in Python Lab	0	0	3	1.5

COURSE OBJECTIVES:

1. To learn about Python programming language syntax, semantics, and the runtime environment.
2. To be familiarized with universal computer programming concepts like data types, containers.
3. To be familiarized with general computer programming concepts like conditional execution, loops & functions.
4. To be familiarized with general coding techniques and object-oriented programming.

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Develop essential programming skills in computer programming concepts like data types, containers	PO1	3
CO2	Apply the basics of programming in the Python language	PO1, PO2	3
CO3	Solve coding tasks related to the fundamental notions and techniques used in object-oriented programming.	PO5	3 3
CO4	Solve coding tasks related OOPS, and Multithreading	PO12	3

LIST OF EXPERIMENTS

S.No.	Name of the experiment	Skill
1.	Exercise – 1 a. Practice Python Installation b. Declaration of Variables, identifiers and type conversions c. Write simple programs by defining variables and assigning values of different basic data types d. Write programs to know data type of a variable using Type statement e. Write programs to do multiple assignments at a time f. Write programs for writing multiple statements in a single line g. Use Input statement, type conversion h. Use different operators in programs	Python installation
2.	Exercise -2 Python programs on Decision Control Statements a. Write programs using selection statements b. Implement programs on and conditional branching statements	Decision Control Statements
3.	Exercises -3 Python programs on looping control structures a. Design and develop programs using Iterative statements- while, for, nested loops b. Use Break, continue, pass statements in programs c. Understand the usage of else statement in loops with a case study	looping control structures
4.	Exercise -4 Identify the need and importance in the creation of Python Functions and Modules a. Write programs for defining and calling functions b. Understand Scope of a variable and Use global statement c. Differentiate fruitful and void functions through a case study d. Apply recursive and Lambda functions e. Understand different kinds of arguments through a case study f. Installing and usage of standard library modules g. Use python packages	Identify the need and importance in the creation of Python Functions and Modules

Program Structure and Detailed Syllabus (VR 20)

5.	<p>Exercise -5 Solve the problems using Strings and understanding the methods and operations on Lists</p> <ol style="list-style-type: none"> Apply string formatting operator Use built in string methods, functions and regular expressions 	Solve the problems using Strings and understanding the methods
6.	<p>Exercise -6 Programs on the implementation of methods and operations of List data structure</p> <ol style="list-style-type: none"> Define a list and write programs to access and modify elements of a list Practice basic list operations, methods Write programs to use list as a stack and queue 	Programs on the implementation of methods and operations of List data structure
7.	<p>Exercise -7 Implement programs to solve the problems using Python other data structures: Tuples and Dictionaries</p> <ol style="list-style-type: none"> Write programs to define a dictionary and write programs to modify values, adding new keys Apply looping over a dictionary Use built in dictionary methods, functions Create a tuple and assign values Use basic tuple operations and comparisons 	Implement programs to solve the problems using Python other data structures: Tuples and Dictionaries
8.	<p>Exercise -8 Implement the Python Classes and Objects to address the real-world scenarios</p> <ol style="list-style-type: none"> Define classes and objects using python for the real-world scenario Defining constructors and using Self Understand public and private members Practice calling class methods from another class Write built in functions to check, get, set and delete attributes 	Implement the Python Classes and Objects to address the real-world scenarios
9.	<p>Exercise -9 Develop the programs to implement parent-child relationship</p> <ol style="list-style-type: none"> Demonstrate different inheritance types Apply polymorphism and method overriding Create abstract classes 	Develop the programs to implement parent-child relationship

Program Structure and Detailed Syllabus (VR 20)

10.	<p>Exercise -10 Write the programs to address the exceptions via exception handling in the development of solutions and implement operator overloading</p> <ol style="list-style-type: none"> Write a simple exception handling program with try-except Write a program for catching multiple exceptions Demonstrate raising and re raising exceptions Apply else and finally clauses Demonstrate the usage of polymorphism in overloading of operators 	Write the programs to address the exceptions via exception handling in the development of solutions and implement operator overloading
11.	<p>Exercise -11</p> <ol style="list-style-type: none"> Create a series from a list, numpy array and dict Convert the index of a series into a column of a dataframe Combine many series to form a dataframe Assign name to the series' index Get the items not common to both series A and series B Get the minimum, 25th percentile, median, 75th, and max of a numeric series Get frequency counts of unique items of a series Bin a numeric series to 10 groups of equal size Find the positions of numbers that are multiples of 3 from a series <p>Get the positions of items of series A in another series B</p>	Pandas
12.	<p>Exercise -12</p> <ol style="list-style-type: none"> create a 1D array Extract items that satisfy a given condition from 1D array Replace items that satisfy a condition without affecting the original array Reshape an array Extract all numbers between a given range from a numpy array Swap two columns in a 2d numpy array Import a dataset with numbers and texts keeping the text intact in python numpy Compute the mean, median, standard deviation of a numpy array Insert values at random positions in an array Find the count of unique values in a numpy array 	numpy

Text Books:

1. Fundamentals of Python First Programs, Kenneth. A. Lambert, Cengage.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson

Reference Books:

1. Introduction to Python Programming, Gowrishankar.S, Veena A, CRC Press.
2. Programming and Problem Solving with Python, Ashok NamdevKamthane, Amit Ashok Kamthane, TMH, 2019.
3. https://www.tutorialspoint.com/python3/python_tutorial.pdf

II Year – I Semester		L	T	P	C
1005202112	Unified Modeling Language Lab	0	0	3	1.5

COURSE OBJECTIVES:

1. To provide a snapshot of the instances in a system and the relationships between the instances.
2. To portray and understand functional requirements of a system

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Create UML Diagrams	PO1	1
		PO2	2
		PO3	3
		PO5	3
		PO11	3
CO2	Create State Chart Diagrams	PO2	3
		PO3	2
		PO5	3
		PO11	3
CO3	Create Interaction Diagrams	PO2	2
		PO3	2
		PO5	3
		PO11	3

LIST OF EXPERIMENTS

S.No.	Name of the experiment	Skill
1.	Demonstration of Rational Rose 98, ARGO UML and IBM RSA tools.	Environment Overview
2.	Draw Class diagram and Use Case Diagram of Library Management System	Design of UML Diagrams
3.	Draw Class diagram and Use Case Diagram of Online Book Shop.	Design of UML Diagrams
4.	Draw Class diagram and Use Case Diagram of Railway Reservation System	Design of UML Diagrams
5.	Draw Class diagram and Use Case Diagram of Banking System	Design of UML Diagrams
6.	Draw Class diagram and Use Case Diagram for Hotel Management system	Design of UML Diagrams
7.	Draw State Chart Diagram for Point Sale System.	Design of UML Diagrams

Program Structure and Detailed Syllabus (VR 20)

8.	Draw State Chart Diagram for Library Management System.	Design of UML Diagrams
9.	Draw State Chart Diagram for Hospital Management System.	Design of UML Diagrams
10.	Draw Interaction Diagrams for Railway Reservation System.	Design of UML Diagrams

Text Books:

1. Software Engineering with UML Book by Bhuvan Unhelkar
2. Object-oriented software engineering Textbook by Bernd Bruegge

Reference Books:

1. The Unified Modeling Language User Guide by Grady Booch et All.

II Year – I Semester		L	T	P	C
1054202180	Competitive Programming	0	0	4	2

COURSE OBJECTIVES:

1. To improve logical and analytical skills
2. To improve programming patterns like recursion

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Apply bit manipulation techniques to solve problems	PO2	3
CO2	Apply the modular programming techniques to simplify the programs.	PO3	2
CO3	Able to solve problems using strings	PO3	3

S.No	Name of the experiment
1	Bit manipulations
2	Number theory: primality
3	Number theory: combinatorics
4	Recursions
5	Arrays
6	String manipulations
7	Time and space complexity optimization
8	Types of errors

Text Books:

1. Problem Solving and Program Design in C, Jeri R. Hanly, Elliot B. Koffman,7th Edition,Pearson.

2. 101 Programming puzzle problems solved: High School Junior to Seniors Join us to win Informatics Olympiad, N.B.Venkateswarlu, Feb, 2015.

Reference Books:

1. Programming in C, PradipDey, Manas Ghosh, 2nd Edition, OxfordUniversityPress.
2. How to Solve it by Computer- R.G.Dromey,PHI.

E-Books:

<https://graphics.stanford.edu/~seander/bithacks.html>

NPTEL/MOOC:

https://onlinecourses.nptel.ac.in/noc21_cs99/preview

II Year – I Semester		L	T	P	C
1000202120	LIFE SKILLS - AUDIT COURSE	2	0	0	0

COURSE OBJECTIVE:

The students will be able to build self-confidence, encourage critical thinking, foster independence and help people to communicate more effectively.

COURSE OUTCOMES:

CO's	At the end of the course, the student will have the ability to:	POs Mapped	Strength of mapping
CO1	Build Self Confidence and Interpersonal and Intrapersonal relationships.	PO12	3
		PO10	3
		PO9	3
CO2	Practice Emotional Competency while communicating with others	PO12	3
		PO10	3
		PO9	3
CO3	Gain Intellectual Competency by practicing ethics and morals	PO12	3
		PO10	3
		PO8	3

UNIT1: LIFE SKILLS: Positive Attitude and Positive Work Ethics, Time Management, Goal Setting: Short term, Long Term. (Activity has to be conducted)

UNIT2: EMOTIONAL INTELLIGENCE: Self Awareness through Johari Window and SWOT analysis (Activity has to be conducted)

UNIT3: PROBLEM SOLVING SKILLS: Critical Thinking and Brain Storming, Creative Thinking, Conflict Management. (Activity has to be conducted)

UNIT4: PUBLIC SPEAKING: Body Language, presentation skills, impromptu presentation, interviewing others. (Activity has to be conducted)

UNIT 5: NPTEL Course/ Coursera /Any relevant Certificate Course has to be done

Assessment: In order to clear internal assessment, the student has to submit Project Report and give Presentation on all the activities he/she has done during the course. The student has to do a certificate course also. (Presentation, Project Report and Certificate in total will be the criteria for the assessment)

References:

Program Structure and Detailed Syllabus (VR 20)

- Barun K. Mitra; (2011), “Personality Development & Soft Skills”, First Edition; Oxford Publishers.
- Kalyana; (2015) “Soft Skill for Managers”; First Edition; Wiley Publishing Ltd.
- Larry James (2016); “The First Book of Life Skills”; First Edition; Embassy Books.
- Shalini Verma (2014); “Development of Life Skills and Professional Practice”; First Edition; Sultan Chand (G/L) & Company
- John C. Maxwell (2014); “The 5 Levels of Leadership”, Centre Street, A division of Hachette Book Group Inc.