Annexure A

CLASS – XI
COMPUTER SCIENCE

Unit-Wise weightage

Part – A External **Marks: 70**

Unit	Topics	Marks
I	Computer System and Organisation	10
II	Computational Thinking and Programing-1	45
III	Society, Law and Ethics	15
Total		**70**

Part – A: External

UNIT 1: Computer System and Organisation **10 marks**

Basic Computer Organisation:

- **Basic Computer Organisation:** Introduction to computer system, Fundamentals, Hardware, Software, Computer System Architecture, Memory, Stored Program Concept, Programming Tools.
- **Number system:** Introduction, Decimal Number System, Binary Numbers, Octal Numbers, Hexadecimal Number System, Octal and Hexadecimal to Binary and Vice-versa.
- **Boolean Algebra and Logic Gates:** Introduction, Boolean Variable, Boolean Operators, Truth Table, Logic Gates, Basic Logic Gates from Universal Gates, Properties of Boolean Algebra, De-Morgan's Laws.
- **Codes:** Introduction, ASCII Code, ISCII Code, Unicode,

Unit II: Computational Thinking and Programming -1 **45 marks**

- **Introduction to Problem Solving:** Introduction, Steps for Problem Solving, Problem Solving Using Decomposition, Algorithm, Algorithms: Definition and Characteristics, Pseudo Code and Flow Chart.
- **Familiarization with the Basics of Python Programming:** Introduction to Python, Features of Python, The Paradigms, Chronology, Installation of Anaconda.
- **Fundamentals of Python:** Introduction, Basic Input Output, Program Execution, The Jupyter Notebook, Value Type and Reference Type, Python Character Set: Tokens, Keywords, Identifiers, Literals and Punctuators, Statements, Comments, Operators, Types and Examples of Operators, Basic Data Types, Errors, Flow of Control.
- **List:** Introduction, Lists, Eval() Function, Accessing Elements: Indexing and Slicing, Mutability, List Operations, Traversing List Using Loops, Making a True Copy of a List, Functions, Illustrations.

- **Tuples:** Introduction, Tuple, Indexing and Slicing, Non-Mutability, Tuple Operations, Traversal, Functions.
- **Dictionary:** Introduction, Associative Arrays and Dictionaries, Displaying Elements of a Dictionary, Important Points Regarding Dictionary, Some Important Built-in Functions of Dictionaries, Input from the User, Making a Shallow Copy of a Dictionary, Illustrations.
- **Strings:** Introduction, Traversing String Using Loops, String Operations, In-built Functions, Illustrations
- **Conditional statements:** Introduction, 'if', if-else, if-elif-else Constructs, The if-elif-else Ladder, Logical Operators, The Ternary Operator, The Get Construct, Examples.
- **Looping:** Introduction, Looping Constructs, Range, Break and Continue, Illustrations, while-else, Nested Loops, Generating Patterns.
- **Introduction to Python Modules:** Introduction, Creating Modules and Packages in Python, Accessing Objects of a Module, Math, Random and Statistics Modules, NumPy.

Unit III: Society, Law and Ethics **15 marks**

- **Society, Law and Ethics-I:** Introduction, Malware, Virus, Worms, Trojans, Spyware, Adware, Cookies, Digital Footprints, Cyber Crime, Cyber Bullying, Identity Thefts, Phishing, Pharming, PC Intrusion, Eavesdropping, Trolling, Spamming, Spreading Rumours, Credit Card Frauds, Cyber Stalking, Illegal Downloading and Digital Piracy, Child Pornography, How to Prevent Computer Viruses, How to Prevent Phishing, How to Prevent Pharming, How to Prevent Eavesdropping, How to Prevent Spamming, Firewall, Authentication, Social Media: Common Networking Sites and Appropriate usage, Social Media Etiquettes, Safely Browsing the Internet, Information Technology Act 2000.
- **Society, Law and Ethics-II:** Introduction, Plagiarism, Intellectual Property, Copyright Infringement, Trademark Infringement, Licensing, Open Movement: Open Data and Open Source, Technology and Society, Education for AII, Electronic Waste Management.

Part – B: Practical **30 Marks**

Sl no	Unit Name	Marks
1.	**Lab Test (12 marks)** Python program (60% logic + 20% documentation + 20% code quality)	12
2.	**Report File + viva (10 marks)** • Report file: Minimum 20 Python programs. • Viva voce	7 3
3.	**Project** (that uses most of the concepts that have been learnt) (See CS-XII for the rules regarding the projects)	8

Suggested Practical list:

Python Programming

- Input a welcome message and display it.
- Input two numbers and display the larger / smaller number.
- Input three numbers and display the largest / smallest number.
- Generate the following patterns using nested loop

Pattern 1	Pattern 2	Pattern 3
*	1 2 3 4 5	A
**	1 2 3 4	AB
***	1 2 3	ABC
****	1 2	ABCD
*****	1	ABCDE

- Write a program to input the value of x and n and print the sum of the following series:
    - $1 + x + x^2 + x^3 + x^4 + \ldots x^n$
    - $1 - x + x^2 - x^3 + x^4 \quad \ldots x^n$
    - $x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots \frac{x^n}{n}$
    - $x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} \cdots \frac{x^n}{n!}$
- Determine whether a number is a perfect number, an armstrong number or a palindrome.
- Input a number and check if the number is a prime or composite number.
- Display the terms of a Fibonacci series.
- Compute the greatest common divisor and least common multiple of two integers.
- Count and display the number of vowels, consonants, uppercase, lowercase characters in string.
- Input a string and determine whether it is a palindrome or not; convert the case of characters in a string.
- Find the largest/smallest number in a list/tuple
- Input a list of numbers and swap elements at the even location with the elements at the odd location.
- Input a list/tuple of elements, search for a given element in the list/tuple.
- Input a list of numbers and find the smallest and largest number from the list.
- Create a dictionary with the roll number, name and marks of n students in a class and display the names of students who have scored marks above 75.

Annexure B

CLASS – XII
COMPUTER SCIENCE

Unit-Wise weightage

Part – A External **Marks: 70**

Unit	**Topics**	**Marks**
I	Computational Thinking and Programming – 2	40
II	Computer Networks	10
III	Database Management	20
Total		**70**

Part – A: External

UNIT I: COMPUTATIONAL THINKING AND PROGRAMMING – 2 40 marks

- **Python Basics I: Fundamentals, Conditional Statements and Looping:** Introduction, Introduction to Python, Value Type and Reference Type, Tokens, Keywords and Identifiers, Statements, Comments, Operators, Types and Examples of Operators, Basic Data Types, Conditional Statements Revisited, Looping.
- **Python Basics II: List, Tuple, Dictionary and Strings:** Introduction, Lists, Tuple, Associate Arrays and Dictionaries, Strings, Selection Sort, Bubble Sort.
- **Functions:** Introduction, Features of a Function, Basic Terminology, Definition and Invocation, Types of Function, Types of Function Based on Parameters and Return Type, Default Values and Positional Parameters, Return Values, Implementing Search, Call by Reference, Scope.
- **Recursion:** Introduction, Recursion Verses Iteration, Factorial, Power, Fibonacci, Binary Search, Greatest Common Divisor.
- **Python Packages:** Introduction, Creating Modules and Packages in Python, Accessing Objects of a Module, Math, Random and Statistics Modules, NumPy.
- **File Handling:** Introduction, The File Handling Mechanism, The Open Function and File Access Modes, Python Functions for File Handling, Implementation and Illustrations.
- **Algorithmic Efficiency:** Introduction, Algorithms: Definition and Characteristics, Notations: Pseudocode, Flow Chart, Strategies for Problem Solving: Recursion versus Iteration. Asymptotic Notation, Complexity, Illustrations, Best-Case, Worst-Case and Average-Case Complexity.
- **Data Visualization:** Introduction, The Plot Function, Plotting Line and Curves, Additional Arguments, Histogram, The Pie Chart, The Bar Chart.
- **Data Structures:** Introduction, Classification of Data Structures, Abstract Data Type, Arrays, Lists, 2-Dimentional List.
- **Stacks and Queues:** Introduction, Stack, Implementation of a Stack, Applications of Stack, Infix to Postfix, Postfix to Infix/Evaluation of Postfix, Queues, Implementation of Queue, Applications of Queues.

UNIT – II: COMPUTER NETWORKS **10 marks**

- **Introduction to Computer Networks:** Communication, Terminology, Modulation, Evolution of Networking, Types of Networks, Cables, Architecture, Network Topology, Devices, OSI Protocol Suite, TCP/IP Protocol Suite, IPV4/Addressing, Address Space, Application layer Protocols, Sending E-Mail, Transport Layer, IP Layer/Routing, Data Link Layer, Network Commands, Mobile Communication Protocols, Miscellaneous Topics.

UNIT – III: DATABASE MANAGEMENT **20 marks**

- **Introduction to Database Management System and MySQL:** Introduction, File System v/s Database Management System, Keys in a Database, Relational Database Management System Exemplified: Employees Database, Relational Database Management System Explained, Examples of Database Management Systems, Introduction to MySQL, Features of MySQL, MySQL Installation, DDL, DCL, DML and TCL, Properties of MySQL.
- **Pre-Defined Functions and Data Description Language:** Introduction, Pre-defined Functions in MySQL, Data Types in MySQL, Creating and Using Database, Creating Table, Describing Table, Insert Values into a Table.
- **Structured Query Language: Revision:** Introduction, Viewing all Tables, Select Statements, Updating Tables, Cartesian Product, Joins, ORDER BY, SQL Functions, GROUP By and HAVING.
- **Interfacing Python:** Introduction, View Existing Databases, Create Database, View Existing Table in a Database, Create Tables, Create Primary Key, Describe Table, ALTER TABLE, Adding Column to a Table, INSERTING Data into a Table, Inserting more than one Record in a Table, SELECT QUERY, ORDER By, DELETE, UPDATE.

Part – B: Practical **30 Marks**

Sl no	Unit Name	Marks
1.	**Lab Test:** • Python program (60% logic + 20% documentation + 20% code quality) • A stub program with Python SQL connectivity must be provided with blanks (4 blanks) to be filled by the student with the desired SQL query.	8 4
2.	**Report File:** Minimum 15 Python programs. • SQL Queries – Minimum 5 sets using one table/two tables • Minimum 4 programs based on Python – SQL connectivity	7
3.	**Project** (using concept learnt in Classes 11 and 12)	8
4.	Viva voce	3

Suggested Practical List:

Python Programming

- Read a text file line by line and display each word separated by a #.
- Read a text file and display the number of vowels/consonants/uppercase/lowercase characters in the file.
- Remove all the lines that contain the character 'a' in a file and write it to another file.
- Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.
- Create a binary file with roll number, name and marks. Input a roll number and update the marks.
- Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).
- Write a Python program to implement a stack using list.
- Create a CSV file by entering user-id and password, read and search the password for given userid.

Database Management

- Create a student table and insert data. Implement the following SQL commands on the student table:
  o ALTER table to add new attributes / modify data type / drop attribute
  o UPDATE table to modify data
  o ORDER By to display data in ascending / descending order
  o DELETE to remove tuple(s)
  o GROUP BY and find the min, max, sum, count and average
- Similar exercise may be framed for other cases.
- Integrate SQL with Python by importing the MySQL module.

C. Project

The aim of the class project is to create something that is tangible and useful using Python file handling/ Python-SQL connectivity. This should be done in groups of two to three students and should be started by students at least 6 months before the submission deadline. The aim here is to find a real world problem that is worthwhile to solve.

Students are encouraged to visit local businesses and ask them about the problems that they are facing. For example, if a business is finding it hard to create invoices for filing GST claims, then students can do a project that takes the raw data (list of transactions), groups the transactions by category, accounts for the GST tax rates, and creates invoices in the appropriate format. Students can be extremely creative here. They can use a wide variety of Python libraries to create user friendly applications such as games, software for their school, software for their disabled fellow students, and mobile applications, of course to do some of these projects, some additional learning is required; this should be encouraged. Students should know how to teach themselves.

The students should be sensitised to avoid plagiarism and violations of copyright issues while working on projects. Teachers should take necessary measures for this.