COMPUTER SCIENCE
Subject Code - 083
Class XI (2026-27)

1. Learning Outcomes

Students should be able to:

a) develop basic computational thinking
b) explain and use data types
c) appreciate the notion of algorithms
d) develop a basic understanding of computer systems- architecture and operating system
e) explain cyber ethics, cyber safety, and cybercrime
f) understand the value of technology in societies along with consideration of gender and disability issues.

2. Distribution of Marks

Unit No.	Unit Name	Marks
1	Computer Systems and Organisation	10
2	Computational Thinking and Programming -1	45
3	Society, Law, and Ethics	15
	Total	**70**

3. Unit wise Syllabus

Unit 1: Computer Systems and Organisation

- Basic computer organisation: Introduction to Computer System, hardware, software, input device, output device, CPU, memory (primary, cache and secondary), units of memory (bit, byte, KB, MB, GB, TB, PB)
- Types of software: System software (Operating systems, system utilities, device drivers), programming tools and language translators (assembler, compiler, and interpreter), application software
- Operating System(OS): functions of the operating system, OS user interface
- Boolean logic: NOT, AND, OR, NAND, NOR, XOR, truth tables and De Morgan's laws, Logic circuits
- Number System: Binary, Octal, Decimal and Hexadecimal number system;

conversion between number systems
- Encoding Schemes: ASCII, ISCII, and Unicode (UTF8, UTF32)

Unit 2: Computational Thinking and Programming - I

- Introduction to Problem-solving: Steps for Problem-solving (Analyzing the problem, developing an algorithm, coding, testing, and debugging), representation of algorithms using flowchart and pseudocode, decomposition
- Familiarization with the basics of Python programming: Introduction to Python, Features of Python, executing a simple “hello world" program, execution modes: interactive mode and script mode, Python character set, Python tokens(keyword, identifier, literal, operator, punctuator), variables, concept of l-value and r-value, use of comments
- Knowledge of data types: Number(integer, floating point,complex), boolean, sequence(string, list, tuple), None, Mapping(dictionary), mutable and immutable data types.
- Operators: arithmetic operators, relational operators, logical operators, assignment operators, augmented assignment operators, identity operators (is, is not), membership operators (in not in)
- Expressions, statement, type conversion, and input/output: precedence of operators, expression, evaluation of an expression, type-conversion (explicit and implicit conversion), accepting data as input from the console and displaying output.
- Errors- syntax errors, logical errors, and run-time errors
- Flow of Control: introduction, use of indentation, sequential flow, conditional and iterative flow
- Conditional statements: if, if-else, if-elif-else, flowcharts, simple programs: e.g.: absolute value, sort 3 numbers and divisibility of a number.
- Iterative Statement: for loop, range(), while loop, flowcharts, break and continue statements, nested loops, suggested programs: generating pattern, summation of series, finding the factorial of a positive number, etc.
- Strings: introduction, string operations (concatenation, repetition, membership and slicing), traversing a string using loops, built-in functions/methods–len(), capitalize(), title(), lower(), upper(), count(), find(), index(), endswith(), startswith(), isalnum(), isalpha(), isdigit(), islower(), isupper(), isspace(),lstrip(), rstrip(), strip(), replace(), join(), partition(), split()
- Lists: introduction, indexing, list operations (concatenation, repetition, membership and slicing), traversing a list using loops, built-in functions/methods–len(), list(), append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), sorted(), min(), max(), sum(); nested lists, suggested programs: finding the maximum, minimum, mean of numeric values stored in a list; linear search on list of numbers and counting the frequency of elements in a list.
- Tuples: introduction, indexing, tuple operations (concatenation, repetition, membership and slicing); built-in functions/methods — len(), tuple(), count(), index(), sorted(), min(), max(), sum(); tuple assignment, nested tuple; suggested programs: finding the minimum, maximum, mean of values stored in a tuple; linear

search on a tuple of numbers, counting the frequency of elements in a tuple.

- Dictionary: introduction, accessing items in a dictionary using keys, mutability of a dictionary (adding a new term, modifying an existing item), traversing a dictionary, built-in functions/methods — len(), dict(), keys(), values(), items(), get(), update(), del, clear(), fromkeys(), copy(), pop(), popitem(), setdefault(), max(), min(), sorted(); Suggested programs: count the number of times a character appears in a given string using a dictionary, create a dictionary with names of employees, their salary and access them.
- Introduction to Python modules: Importing module using 'import <module>' and using from statement, importing math module (pi, e, sqrt(), ceil(), floor(), pow(), fabs(), sin(), cos(), tan()); random module (random(), randint(), randrange()), statistics module (mean(), median(), mode()).

Unit 3: Society, Law and Ethics

- Digital Footprints
- Digital Society and Netizen: net etiquettes, communication etiquettes, social media etiquettes
- Data Protection: Intellectual property rights (copyright, patent, trademark), violation of IPR (plagiarism, copyright infringement, trademark infringement), open source software and licensing (Creative Commons, GPL and Apache)
- Cyber Crime: definition, hacking, eavesdropping, phishing and fraud emails, ransomware, cyber trolls, cyber bullying
- Cyber safety: safely browsing the web, identity protection, confidentiality
- Malware: viruses, trojans, adware
- E-waste management: proper disposal of used electronic gadgets.
- Information Technology Act (IT Act)
- Technology and society: Gender and disability issues while teaching and using computers

4. Practical

S.No.	Unit Name	Marks (Total=30)
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	**7**
	Viva voce	**3**
3.	Project (that uses most of the concepts that have been learnt)	**8**

5. 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 loops:

Pattern-1	Pattern-2	Pattern-3
* ** *** **** *****	12345 1234 123 12 1	A AB ABC ABCD ABCDE

- Write a program to input the value of x and n and print the sum of the following series:
  - $1 + x + x^2 + x^3 + x^4 + \cdots x^n$
  - $1 - x + x^2 - x^3 + x^4 - \cdots x^n$
  - $x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \cdots \frac{x^n}{n}$
  - $x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \frac{x^n}{n!}$
- 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.
- Create a dictionary with the roll number, name and marks of n students in a class and display the names of students who have marks above 75.

6. Suggested Reading Material

- NCERT Textbook for Computer Science (Class XI)
- Support Material on CBSE website

COMPUTER SCIENCE
Subject Code – 083
Class XII (2026-27)

1. Prerequisites

Computer Science- Class XI

2. Learning Outcomes

Student should be able to

a) apply the concept of function.

b) explain and use the concept of file handling.

c) use basic data structure: Stacks

d) explain basics of computer networks.

e) use Database concepts, SQL along with connectivity between Python and SQL.

3. Distribution of Marks:

Unit No.	Unit Name	Marks
1	Computational Thinking and Programming — 2	40
2	Computer Networks	10
3	Database Management	20
	Total	**70**

4. Unit wise Syllabus

Unit 1: Computational Thinking and Programming – 2

- Revision of Python topics covered in Class XI.
- Functions: types of function (built-in functions, functions defined in module, user defined functions), creating user defined function, arguments and parameters, default parameters, positional parameters, function returning value(s), flow of execution, scope of a variable (global scope, local scope)
- Exception Handling: Introduction, handling exceptions using try-except-finally blocks
- Introduction to files, types of files (Text file, Binary file, CSV file), relative and absolute paths

- Text file: opening a text file, text file open modes (r, r+, w, w+, a, a+), closing a text file, opening a file using with clause, writing/appending data to a text file using write() and writelines(), reading from a text file using read(), readline() and readlines(), seek and tell methods, manipulation of data in a text file
- Binary file: basic operations on a binary file: open using file open modes (rb, rb+, wb, wb+, ab, ab+), close a binary file, import pickle module, dump() and load() method, read, write/create, search, append and update operations in a binary file
- CSV file: import csv module, open / close csv file, write into a csv file using writer(),writerow(),writerows() and read from a csv file using reader()
- Data Structure: Stack, operations on stack (push & pop), implementation of stack using list.

Unit 2: Computer Networks

- Evolution of networking: introduction to computer networks, evolution of networking (ARPANET, NSFNET, INTERNET)
- Data communication terminologies: concept of communication, components of data communication (sender,receiver, message, communication media, protocols), measuring capacity of communication media (bandwidth, data transfer rate), IP address, switching techniques (Circuit switching, Packet switching)
- Transmission media: Wired communication media (Twisted pair cable, Co-axial cable, Fiber-optic cable), Wireless media (Radio waves, Micro waves, Infrared waves)
- Network devices (Modem, Ethernet card, RJ45, Repeater, Hub, Switch, Router, Gateway, WIFI card)
- Network topologies and Network types: types of networks (PAN, LAN, MAN, WAN), networking topologies (Bus, Star, Tree)
- Network protocol: HTTP, FTP, PPP, SMTP, TCP/IP, POP3, HTTPS, TELNET, VoIP
- Introduction to web services: WWW, Hyper Text Markup Language (HTML), Extensible Markup Language (XML), domain names, URL, website, web browser, web servers, web hosting

Unit 3: Database Management

- Database concepts: introduction to database concepts and its need
- Relational data model: relation, attribute, tuple, domain, degree, cardinality, keys (candidate key, primary key, alternate key, foreign key)
- Structured Query Language: introduction, Data Definition Language and Data Manipulation Language, data type (char(n), varchar(n), int, float, date), constraints (not null, unique, primary key), create database, use database, show databases, drop database, show tables, create table, describe table, alter table (add and remove an attribute, add and remove primary key), drop table, insert, delete, select, operators (mathematical, relational and logical), aliasing, distinct clause, where clause, in, between, order by, meaning of null, is null, is not null, like, update command, delete command, aggregate functions (max, min, avg, sum, count), group by, having clause, joins: cartesian product on two tables, equi-join and natural join
- Interface of python with an SQL database: connecting SQL with Python, performing

insert, update, delete queries using cursor, display data by using connect(), cursor(), execute(), commit(), fetchone(), fetchall(), rowcount, creating database connectivity applications, use of %s format specifier or format() to perform queries

5. Practical

S.No	Unit Name	Marks (Total=30)
1	Lab Test: 1. Python program (60% logic + 20% documentation + 20% code quality)	8
	2. SQL queries (4 queries based on one or two tables)	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 concepts learnt in Classes 11 and 12)	8
4	Viva voce	3

6. 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:
    - ALTER table to add new attributes / modify data type / drop attribute
    - UPDATE table to modify data
    - ORDER By to display data in ascending / descending order
    - DELETE to remove tuple(s)
    - 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 suitable module.

7. Suggested Reading Material

- NCERT Textbook for COMPUTER SCIENCE (Class XII)
- Support Materials on the CBSE website.

8. 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 sensitized to avoid plagiarism and violations of copyright issues while working on projects. Teachers should take necessary measures for this.