

# Computer Science (New)

CLASS-XI

Code No. 083

(Optional for the academic year 2018-19 and mandatory for the academic year 2019-20 onwards)

## 1. Prerequisites

No major prerequisites are required for this course other than basic Mathematical skills. However, it will be helpful if the student has a basic knowledge of Computer Applications.

## 2. Learning Outcomes

1. Develop basic computational thinking. Learn how to reason with variables, state transitions, conditionals, and iteration.
2. Understand the notion of data types, and higher order data structures such as lists, tuples, and dictionaries.
3. Appreciate the notion of an algorithm, and understand its structure, including how algorithms handle corner cases.
4. Develop a basic understanding of computer systems - architecture, OS, mobile and cloud computing.
5. Learn basic SQL programming.
6. Learn all about cyber safety.

## 3. Distribution of Marks

Unit No.	Unit Name	Marks
1.	Programming and Computational Thinking - 1	35
2.	Computer Systems and Organisation	10
3.	Data Management - 1	15
4.	Society, Law and Ethics - 1	10
5.	Practical	30
	Total	100

### 4.1 Unit 1: Programming and Computational Thinking (PCT-1) (80 Theory + 70 Practical)

- Familiarization with the basics of Python programming: a simple "hello world" program, process of writing a program, running it, and print statements; simple data-types: integer, float, string
- Introduce the notion of a variable, and methods to manipulate it (concept of L-value and R-value even if not taught explicitly)
- Knowledge of data types and operators: accepting input from the console, assignment statement, expressions, operators and their precedence.
- Conditional statements: if, if-else, if-elif-else; simple programs: e.g.: absolute value, sort 3 numbers, and divisibility.
- Notion of iterative computation and control flow: for, while, flowcharts, decision trees and pseudo code; write a lot of programs: interest calculation, primarily testing, and factorials.
- Idea of debugging: errors and exceptions; debugging: pdb, break points.
- Lists, tuples and dictionary: finding the maximum, minimum, mean; linear search on list/tuple of numbers, and counting the frequency of elements in a list using a dictionary. Introduce the notion of accessing elements in a collection using numbers and names.
- Sorting algorithm: bubble and insertion sort; count the number of operations while sorting.
- Strings: compare, concat, substring; notion of states and transitions using state transition diagrams.

**4.2. Unit 2: Computer Systems and Organisation (CSO)****(20 Theory + 6 Practical)**

- Basic computer organisation: description of a computer system and mobile system, CPU, memory, hard disk, I/O, battery, power.
- Types of software: application, OS, utility, libraries.
- Language of Bits: bit, byte, MB, GB, TB, and PB.
- Boolean logic: OR, AND, NAND, NOR, XOR, NOT, truth tables, De Morgan's laws
- Information representation: numbers in base 2, 8, 16, unsigned integers, binary addition
- Strings: ASCII, UTF8, UTF32, ISCII (Indian script code)
- Execution of a program: basic flow of compilation – program → binary → execution
- Interpreters (process one line at a time), difference between a compiler and an interpreter
- Running a program: Notion of an operating system, how an operating system runs a program, idea of loading, operating system as a resource manager.
- Concept of cloud computers, cloud storage (public/private), and brief introduction to parallel computing.

**4.3. Unit 3: Data Management (DM-1)****(30 Theory+ 24 Practical)**

- Relational databases: idea of a database and the need for it, relations, keys, primary key, foreign key; use SQL commands to create a table, keys, foreign keys; insert/delete an entry, delete a table.
- SQL commands: select, project, and join; indexes, and a lot of in-class practice.
- Basics of NoSQL databases - Mongo DB.

**4.4. Unit 4: Society, Law and Ethics (SLE-1) - Cyber safety****(10 Theory)**

- Cyber safety: safely browsing the web, identity protection, confidentiality, social networks, cyber trolls and bullying
- Appropriate usage of social networks: spread of rumours, and common social networking sites (Twitter, LinkedIn, and Facebook) and specific usage rules.
- Safely accessing web sites: adware, malware, viruses, Trojans
- Safely communicating data: secure connections, eavesdropping, phishing and identity verification.

**5. Practical**

S.No.	Unit Name	Marks (Total=30)
1.	<b>Lab Test (12 marks)</b>	
	Python program (60% logic + 20% documentation + 20% code quality)	<b>8</b>
	SQL program (at least 4 queries)	<b>4</b>
2.	<b>Report File + viva (10 marks)</b>	
	Report file: Minimum 20 Python programs and 8 SQL commands	<b>7</b>
	Viva voce (based on the report file)	<b>3</b>
3.	Project (that uses most of the concepts that have been learnt) (See CS-XII for the rules regarding the projects).	<b>8</b>

**5.1. Programming in Python:** At least the following Python concepts should be covered in the lab sessions: expressions, conditionals, loops, list, dictionary, and strings. The following are some representative lab assignments.

- Find the largest and smallest numbers in a list.
- Find the third largest number in a list.
- Test for primality.
- Find whether a string is a palindrome or not.
- Given two integers  $x$  and  $n$ , compute  $x^n$ .
- Compute the greatest common divisor and the least common multiple of two integers.
- Test if a number is equal to the sum of the cubes of its digits. Find the smallest and largest such numbers.

**5.2. Data Management: SQL Commands** At least the following SQL commands should be covered during the labs: create, insert, delete, select, and join. The following are some representative assignments.

- Create a student table with the student id, name, and marks as attributes where the student id is the primary key.
- Insert the details of a new student in the above table.
- Delete the details of a particular student in the above table.
- Use the select command to get the details of the students with marks more than 80.
- Create a new table (name, date of birth) by joining two tables (student id, name) and (student id, date of birth).
- Create a new table (order ID, customer Name, and order Date) by joining two tables (order ID, customer ID, and order Date) and (customer ID, customer Name, contact Name, country).

# Computer Science

CLASS-XII  
Code No. 083

Optional for the academic year 2019-20 and mandatory for the academic year 2020-21 onwards

## 1. Prerequisites

Computer Science- Class XI

## 2. Learning Outcomes

1. Understand the concept of functions and recursion.
2. Learn how to create and use Python libraries.
3. Learn file handling.
4. Learn about the concept of efficiency in algorithms and computing in general.
5. Learn basic data structures: lists, stacks, and queues.
6. Get a basic understanding of computer networks: network stack, basic network hardware, basic protocols, and basic tools.
7. Connect a Python program with an SQL database, and learn aggregation functions in SQL.
8. Have a clear understanding of cyber ethics and cybercrime. Understand the value of technology in societies, gender and disability issues, and the technology behind biometric ids.

## 3. Distribution of Marks

Unit No.	Unit Name	Marks
1.	Programming and Computational Thinking – 2	30
2.	Computer Networks	15
3.	Data Management – 2	15
4.	Society, Law and Ethics – 2	10
5.	Practicals	30
	Total	100

### 4.1 Unit 1: Programming and Computational Thinking (PCT-2) (80 Theory + 70 Practical)

- Revision of the basics of Python
- Functions: scope, parameter passing, mutable/immutable properties of data objects, pass arrays to functions, return values, functions using libraries: mathematical, and string functions.
- File handling: open and close a file, read, write, and append to a file, standard input, output, and error streams, relative and absolute paths.
- Using Python libraries: create and import Python libraries
- Recursion: simple algorithms with recursion: factorial, Fibonacci numbers; recursion on arrays: binary search
- Idea of efficiency: performance defined as inversely proportional to the wall clock time, count the number of operations a piece of code is performing, and measure the time taken by a program. Example: take two different programs for the same problem, and understand how the efficient one takes less time.
- Data visualization using Pyplot: line chart, pie chart, and bar chart.
- Data-structures: lists, stacks, queues.

#### 4.2 Unit 2: Computer Networks (CN)

(30 Theory + 10 Practical)

- Structure of a network: Types of networks: local area and wide area (web and internet), new technologies such as cloud and IoT, public vs. private cloud, wired and wireless networks; concept of a client and server.
- Network devices such as a NIC, switch, hub, router, and access point.
- Network stack: amplitude and frequency modulation, collision in wireless networks, error checking, and the notion of a MAC address, main idea of routing. IP addresses: (v4 and v6), routing table, router, DNS, and web URLs, TCP: basic idea of retransmission, and rate modulation when there is congestion (analogy to a road network), Protocols: 2G, 3G, 4G, Wi-Fi. What makes a protocol have a higher bandwidth?
- Basic network tools: traceroute, ping, ipconfig, nslookup, whois, speed-test.
- Application layer: HTTP (basic idea), working of email, secure communication: encryption and certificates (HTTPS), network applications: remote desktop, remote login, HTTP, FTP, SCP, SSH, POP/IMAP, SMTP, VoIP, NFC.

#### 4.3 Unit 3: Data Management (DM-2)

(20 Theory + 20 Practical)

- Write a minimal Django based web application that parses a GET and POST request, and writes the fields to a file - flat file and CSV file.
- Interface Python with an SQL database
- SQL commands: aggregation functions – having, group by, order by.

#### 4.4. Unit 4: Society, Law and Ethics (SLE-2)

(10 Theory)

- Intellectual property rights, plagiarism, digital rights management, and licensing (Creative Commons, GPL and Apache), open source, open data, privacy.
- Privacy laws, fraud; cyber-crime- phishing, illegal downloads, child pornography, scams; cyber forensics, IT Act, 2000.
- Technology and society: understanding of societal issues and cultural changes induced by technology.
- E-waste management: proper disposal of used electronic gadgets.
- Identity theft, unique ids, and biometrics.
- Gender and disability issues while teaching and using computers.

#### 5. Practical

S.No.	Unit Name	Marks (Total=30)
1.	<b>Lab Test (10 marks)</b>	
	Python program (60% logic + 20% documentation + 20% code quality)	7
	Small Python program that sends a SQL query to a database and displays the result. A stub program can be provided.	3
2.	<b>Report File + viva(9 marks)</b>	
	Report file: Minimum 21 Python programs. Out of this at least 4 programs should send SQL commands to a database and retrieve the result; at least 1 program should implement the web server to write user data to a CSV file.	7
	Viva voce based on the report file	2
3.	<b>Project + viva (11 marks)</b>	
	Project (that uses most of the concepts that have been learnt)	8
	Project viva voce	3

Some sample lab assignments are as follows:

### 5.1. Programming in Python:

- Recursively find the factorial of a natural number.
- Read a file line by line and print it.
- Remove all the lines that contain the character `a` in a file and write it to another file.
- Write a Python function  $\sin(x, n)$  to calculate the value of  $\sin(x)$  using its Taylor series expansion up to  $n$  terms. Compare the values of  $\sin(x)$  for different values of  $n$  with the correct value.
- Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).
- Write a recursive code to find the sum of all elements of a list.
- Write a recursive code to compute the  $n^{\text{th}}$  Fibonacci number.
- Write a Python program to implement a stack and queue using a list data-structure.
- Write a recursive Python program to test if a string is a palindrome or not.
- Write a Python program to plot the function  $y = x^2$  using the pyplot or matplotlib libraries.
- Create a graphical application that accepts user inputs, performs some operation on them, and then writes the output on the screen. For example, write a small calculator. Use the tkinter library.
- Open a webpage using the urllib library.
- Compute EMIs for a loan using the numpy or scipy libraries.
- Take a sample of 10 phishing e-mails and find the most common words.

### 5.2. Data Management: SQL and web-server

- Find the min, max, sum, and average of the marks in a student marks table.
- Find the total number of customers from each country in the table (customer ID, customer name, country) using group by.
- Write a SQL query to order the (student ID, marks) table in descending order of the marks.
- Integrate SQL with Python by importing the MySQL module
- Write a Django based web server to parse a user request (POST), and write it to a CSV file.

## 6. Project

The aim of the class project is to create something that is tangible and useful. This should be done in groups of 2 to 3 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 this projects, some additional learning is required; this should be encouraged. Students should know how to teach themselves.

If three people work on a project for 6 months, at least 500 lines of code is expected. The committee has also been made aware about the degree of plagiarism in such projects. Teachers should take a very strict look at this situation, and take very strict disciplinary action against students who are cheating on lab assignments, or projects, or using pirated software to do the same. Everything that is proposed can be achieved using absolutely free, and legitimate open source software.