Computer Applications - 1 (Class 9)
Effective from the 2018-19 academic year.

1. Prerequisites

No background in computer science is required.

2. Learning Outcomes

1. Familiarity with basics of computers.
2. Ability to navigate the file system.
3. Create and edit rich text documents, spreadsheets, and presentations.
4. Perform basic data manipulation using spreadsheets.
5. Use Indian languages in documents.
6. Send and receive emails, follow email etiquette, and communicate over the internet.
7. Create and upload videos.
8. Safe and correct usage of websites, social networks, chat sites, and email.

3. Distribution of Marks

Unit No.	Unit Name	Marks
1.	Basics of Information Technology	5
2.	Cyber safety	10
3.	Office Tools	5
4.	Scratch/Python	10
5.	Lab Exercises	70
	Total	100

4.1. Unit 1: Basics of Information Technology

- Familiarity with the basics of computers: design of computers, and overview of communication technologies
- Computer Systems: characteristics of a computer, components of a computer system – CPU, memory, storage devices and I/O devices
- Memory: primary (RAM and ROM) and secondary memory
- Storage devices: hard disk, CD ROM, DVD, pen/flash drive, memory stick
- I/O devices: keyboard, mouse, monitor, printer, scanner, web camera
- Types of software: system software (operating systems), application software, mobile applications
- Operating systems: kernel, device drivers, and file systems (very basic idea)
- Computer networking: wired/wireless communication, common protocols: Wi-Fi, Bluetooth, cloud computers (private/public)
- Multimedia: images, audio, video, animation
- Chat sites, and social networks.

4.2. Unit 2: Cyber-safety

- Safely browsing the web and using social networks: identity protection, proper usage of passwords, privacy, confidentiality of information, cyber stalking, reporting cybercrimes
- Safely accessing websites: viruses and malware

4.3. Unit 3: Office tools

- Introduction to a word processor: create and save a document.
- Edit and format text: text style (B, I, U), font type, font size, text colour, alignment of text. Format paragraphs with line and/or paragraph spacing. Add headers and footers, numbering pages, grammar and spell check utilities, subscript and superscript, insert symbols, use print preview, and print a document.

- Insert pictures, change the page setting, add bullets and numbering, borders and shading, and insert tables – insert/delete rows and columns, merge and split cells.
- Use auto-format, track changes, review comments, use of drawing tools, shapes and mathematical symbols.
- Presentation tool: understand the concept of slide shows, basic elements of a slide, different types of slide layouts, create and save a presentation, and learn about the different views of a slide set – normal view, slide sorter view and hand-outs.
- Edit and format a slide: add titles, subtitles, text, background, and watermark, headers and footers, and slide numbers.
- Insert pictures from files, create animations, add sound effects, and rehearse timings.
- Spreadsheets: concept of a worksheet and a workbook, create and save a worksheet.
- Working with a spreadsheet: enter numbers, text, date/time, series using auto fill; edit and format a worksheet including changing the colour, size, font, alignment of text; insert and delete cells, rows and columns. Enter a formula using the operators (+,-,*, /), refer to cells, and print a worksheet.
- Use simple statistical functions: SUM (), AVERAGE (), MAX (), MIN (), IF () (without compound statements); embed charts of various types: line, pie, scatter, bar and area in a worksheet.

4.4. Unit 4: Scratch or Python

Alternative 1: Educational programming language - Scratch

- Introduction to Scratch.
- Drag and drop commands, creating simple scripts, repeating blocks of commands.
- Discuss x-y plane, create scripts to move the cat (Scratch mascot).
- Create a script to draw diagrams using the pen feature.

Alternative 2: Python - (provided as an option to children with special needs)

- Introduction to Python
- A simple “Hello World" program
- Running a Python program
- The notion of data-types and variables: integer, float, string
- Arithmetic operations: +, -, *, /

5. Lab Exercises

- Basic I/O devices: use the mouse and keyboard, draw a figure.
- Working with the operating system: Navigation of the file system using a mouse and keyboard, and then doing the same with shell commands.
- Word processing: create a text document, create a letter, report, and greeting card.
- Create a text document with figures in it. It should describe a concept taught in another course.
- Discuss the following in a text document about the basic organisation of a computer: CPU, memory, input/output devices, hard disk.
- Create a text document in an Indian language other than English.
- Create a presentation.
- Create a presentation with animation.
- Create and edit existing images, and then include them in a presentation.
- Animate pictures and text with sound effects in a presentation
- Create a simple spreadsheet and perform the following operations: min, max, sum, and average.
- Create different types of charts using a spreadsheet: line, bar, and pie.
- Send an email to your friends. Attach some documents that you have prepared earlier. Put some friend in the CC and BCC list. Interact with friends to find out who was in the BCC list.
- Do an online chat with multiple friends. Transmit documents using the chat platform.
- Create a video and upload it on YouTube.
- Write basic Scratch/Python programs.

Breakup of marks for the practicals:

S.No.	**Unit Name**	**Marks**
1.	Lab Test (30 marks)	
	Proficiency with the OS	2.5
	Word processing	5
	Handling spreadsheets	7.5
	Creating presentations	7.5
	Writing basic Python/Scratch programs	7.5
2.	Report File + viva (25 marks)	
	Report file:4 documents each with a word processor, spreadsheet, and presentation tool	20
	Viva voce (based on the report file)	5
3.	Project (that uses most of the concepts that have been learnt)	15

Computer Applications - 2 (Class 10)
Effective from the 2019-20 academic year.

1. Prerequisites

Computer Applications – 1

2. Learning Outcomes

1. Create a simple website
2. Embed images, audio and video in an HTML page
3. Use style sheets to beautify the web pages.
4. Write iterative programs with Scratch/Python.
5. Interface a web site with a web server and record the details of a user's request.
6. Knowledge of basic cyber ethics

3. Distribution of Marks

Unit No.	Unit Name	Marks
1.	Networking	5
2.	HTML	20
3.	Cyber ethics	5
4.	Scratch/Python Theory	10
5.	Practicals	60
	Total	100

4.1. Unit 1: Networking

- Internet: world wide web, web servers, web clients, web sites, web pages, web browsers, blogs, news groups, HTML, web address, e-mail address, downloading and uploading files from a remote site. Internet protocols: HTTP, HTTPS. Remote login and file transfer protocols: SSH, SFTP, FTP, SCP, TELNET.
- Services available on the internet: information retrieval, locating sites using search engines and finding people on the net;
- Web services: chat, email, video conferencing, e-Learning, e-Banking, e-Shopping, e-Reservation, e-Governance, e-Groups, social networking.
- Mobile technologies: SMS, MMS, 3G, 4G.

4.2. Unit 2: HTML

- Introduction to web page designing using HTML: create and save an HTML document, access a web page using a web browser.
- HTML tags: html, head, title, body, *br (break), hr(horizontal rule), inserting comments, h1..h6 (heading), p (paragraph), b (bold), i (italics), u (underline), ul (unordered list), ol (ordered list), and li (list item). Description lists: dl, dt and dd.*
- Insert images: img (attributes: src, width, height, alt), sup (super script), sub (subscript).
- Create a table using the tags: table, tr, th, td, rowspan, colspan
- Links: significance of linking, anchor element (attributes: href, mailto), targets.
- Cascading style sheets: colour, background-colour, border-style, margin, height, width, outline, font (family, style, size), align, float.

4.3. Unit 3: Cyber ethics

- E-commerce: privacy, fraud, secure data transmission
- Intellectual property rights, plagiarism, and digital property rights
- Software licenses and the open source software movement
- Freedom of information and the digital divide

4.4. Unit 4: Scratch or Python (Theory and Practical)

Alternative 1: Scratch

- Revision of the basics of Scratch
- Tempo, variables, and events
- Coordinates and conditionals
- Drawing with iteration
- Update variables repeatedly, iterative development, ask and answer blocks
- Create games, animated images, stories and songs

Alternative 2: Python (only for children with special needs)

- Revision of Python basics
- Conditionals: if, if-else statements
- Loops: for, while (e.g., sum of first 10 natural numbers)
- Practice simple programs

5. Lab Exercises

- Create static web pages.
- Use style sheets to enforce a format in an HTML page (CSS).
- Embed pictures, audio and videos in an HTML page.
- Add tables and frames in an HTML page.
- Decorate web pages using graphical elements.
- Create a website using several webpages. Students may use any open source or proprietary tool.
- Work with HTML forms: text box, radio buttons, checkbox, password, list, combo box.
- Install a web server (Django) and send a GET/POST request to the server. The server will append the fields in the request to a CSV file. The code of the server will be downloadable from CBSE's website.
- Open this CSV file using a spreadsheet.
- Write a blog using HTML pages discussing: viruses, malware, spam and antiviruses
- Do a small project using HTML forms and a Django web server (strictly NO dbms).
- Create a web page discussing plagiarism. List some reported cases of plagiarism and the consequent punishment meted out. Explain the nature of the punishment in different countries as per their IP laws.
- Create simple stories with Scratch (involving at least two objects/characters) and iteration OR write programs for finding the sum/product of first *n* natural numbers using Python

Breakup of marks for the practicals:

S.No.	Unit Name	Marks
1.	Lab Test (30 marks)	
	HTML (design one web page based on a diagram)	15
	Scratch or Python (write one program)	15
2.	Report File + viva (20 marks)	
	Report file: At least 8 HTML pages, and at least 7 Scratch/Python programs.	15
	Viva voce (based on the report file)	5
3.	Project (that uses most of the concepts that have been learnt)	10

Computer Science - 1 (Class 11)

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

1. Prerequisites

Since a lot of students join CBSE schools from schools run by a state board, we are not assuming any prerequisites for this course other than basic mathematical skills. However, it will be helpful if the student has a basic knowledge of Computer Applications 1 and 2.

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
1.	Lab Test (12 marks)	
	Python program (60% logic + 20% documentation + 20% code quality)	8
	SQL program (at least 4 queries)	4
2.	Report File + viva (10 marks)	
	Report file: Minimum 20 Python programs and 8 SQL commands	7
	Viva voce (based on the report file)	3
3.	Project (that uses most of the concepts that have been learnt) See CS-2 regarding the rules regarding the projects.	8

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 primarily.
- 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 - 2 (Class 12)

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

1. Prerequisites

Computer Science - 1

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
1.	Lab Test (10 marks)	
	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.	Report File + viva(9 marks)	
	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
	Project + viva (11 marks)	
3.	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^{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.

Informatics Practices (IP) – 1 (Class 11)

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

1. Prerequisites

Since a lot of students join CBSE schools from schools run by a state board, we are not assuming any pre-requisites for this course other than basic mathematical skills. However, it will be helpful if the student has a basic knowledge of Computer Applications 1 and 2.

2. Learning Outcomes

1. Basic computational thinking. Learn how to reason with variables, state transitions, conditionals, and iteration.
2. Notion of data types, and higher order data structures such as lists, and dictionaries.
3. Concepts of data handling: creating, managing and working with Data Frames using Python Pandas.
4. Structure of simple SQL queries.
5. Cyber safety.

3. Distribution of Marks

Unit No.	Unit Name	Marks
1.	Programming and Computational Thinking	30
2.	Data Handling - 1	20
3.	Data Management - 1	10
4.	Society, Law and Ethics - 1	10
5.	Practicals	30
	Total	100

4.1 Unit 1: Programming and Computational Thinking (PCT-1) (70 Theory + 60 Practical)

- Basic computer organisation: describe a computer system and mobile system, CPU, memory, hard disk, I/O, battery, power, transition from a calculator to a computer
- 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, divisibility.
- Notion of iterative computation and control flow: for, while, flowcharts, decision trees and pseudo code; write a lot of programs: interest calculation, EMI, tax calculation (examples from GST), standard deviation, correlation
- Lists and dictionary: finding the maximum, minimum, mean; linear search on a list of numbers, and counting the frequency of elements in a list using a dictionary.
- Text handling: compare, concat, and substring operations.
- Introduction to Python modules: creating and importing.

4.2 Unit 2: Data Handling (DH-1) (30 Theory + 20 Practical)

4.2.1. Introduction to Python Pandas

- Introduction to data structures in Pandas: Series, and Data Frame
- Operations on a Series: head, tail, vector operations
- Data Frame operations: create, display, iteration, select column, add column, delete column
- Binary operations in a Data Frame: add, sub, mul, div, radd, rsub
- Matching and broadcasting operations
- Missing data and filling values.
- Comparisons, Boolean reductions, comparing Series, and combining Data Frames.

4.2.2. Transfer data between CSV files/SQL databases, and Data Frame objects.

4.3. Unit 3: Data Management (DM-1) (30 Theory + 20 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, and foreign keys; insert/delete an entry, delete a table.
- Basic SQL: select, project, and join; indexes, and a lot of in-class practice.

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, and phishing and identity verification.

5. Practical

S.No.	Unit Name	Marks
1.	Lab Test (12 marks)	
	Python programs to test PCT (60% logic + 20% documentation + 20% code quality)	4
	Python programs to test data handling (same rules as above)	4
	SQL program (at least 4 queries)	4
2.	Report File + viva (10 marks)	
	Report file: Minimum 20 Python programs (PCT + DH) and at least 8 SQL commands	7
	Viva voce (based on the report file)	3
3.	Project (that uses most of the concepts that have been learnt) See IP-2 for the rules regarding the projects.	8

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.
- Find the sum of squares of the first 100 natural numbers.
- 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 in the range of 100 to 1000.

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).

5.3. Data Handling: The following are some representative lab assignments.

- Subtract the mean of a row from each element of the row in a Data Frame.
- Filter out rows based on different criteria such as redundant rows (same data as the row above or below).
- Find the sum of each column, or find the column with the lowest mean.
- Locate the 3 largest values in a data frame.
- Replace all negative values in a data frame with a 0.

Informatics Practices (IP) - 2 (Class 12)

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

1. Prerequisites

Informatics Practices - 1

2. Learning Outcomes

1. Understand aggregation operations, descriptive statistics, and re-indexing columns in a Data Frame.
2. Apply functions row-wise and element-wise on a Data Frame.
3. Understand basic software engineering: models, activities, business use-case diagrams, and version control systems.
4. Connect a Python program with a SQL database, and learn aggregation functions in SQL.
5. 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.	Data Handling - 2	30
2.	Basic Software Engineering	15
3.	Data Management - 2	15
4.	Society, Law and Ethics - 2	10
5.	Practicals	30
	Total	100

4.1. Unit 1: Data Handling (DH-2) (80 Theory + 70 Practical)

4.1.1. Python Pandas

- Advanced operations on Data Frames: pivoting, sorting, and aggregation
- Descriptive statistics: min, max, mode, mean, count, sum, median, quartile, var
- Create a histogram, and quantiles.
- Function application: pipe, apply, aggregation (group by), transform, and apply map.
- Reindexing, and altering labels.

4.1.2. Numpy

- 1D array, 2D array
- Arrays: slices, joins, and subsets
- Arithmetic operations on 2D arrays
- Covariance, correlation and linear regression

4.1.3. Plotting with Pyplot

- Plot bar graphs, histograms, frequency polygons, box plots, and scatter plots.

4.2 Unit 2: Basic Software Engineering (BSE) (25 Theory + 10 Practical)

- Introduction to software engineering

- Software Processes: waterfall model, evolutionary model, and component based model
- Delivery models: incremental delivery, spiral delivery
- Process activities: specification, design/implementation, validation, evolution
- Agile methods: pair programming, and Scrum
- Business use-case diagrams
- Practical aspects: Version control system (GIT), and do case studies of software systems and build use-case diagrams

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) (15 Theory)

- Intellectual property rights, plagiarism, digital rights management, and licensing (Creative Commons, GPL and Apache), open source, open data, privacy.
- Privacy laws, fraud; cybercrime- 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.
- Role of new media in society: online campaigns, crowdsourcing, smart mobs
- Issues with the internet: internet as an echo chamber, net neutrality, internet addiction
- Case studies - Arab Spring, WikiLeaks, Bit coin

5. Practical

S.No.	Unit Name	Marks
1.	Lab Test (10 marks)	
	Python programs for data handling (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.	Report File + viva(9 marks)	
	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
	Project + viva (11 marks)	
3.	Project (that uses most of the concepts that have been learnt)	8
	Project viva voce	3

5.1. Data Management: SQL+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 MYSQL dB
- Write a Django based web server to parse a user request (POST), and write it to a CSV file.

5.2. Data handling using Python libraries

- Use map functions to convert all negative numbers in a Data Frame to the mean of all the numbers.
- Consider a Data Frame, where each row contains the item category, item name, and expenditure.
    - Group the rows by the category, and print the total expenditure per category.
- Given a Series, print all the elements that are above the 75th percentile.
- Given a day’s worth of stock market data, aggregate it. Print the highest, lowest, and closing prices of each stock.
- Given sample data, plot a linear regression line.
- Take data from government web sites, aggregate and summarize it. Then plot it using different plotting functions of the PyPlot library.

5.3. Basic Software Engineering

- Business use-case diagrams for an airline ticket booking system, train reservation system, stock exchange
- Collaboratively write a program and manage the code with a version control system (GIT)

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.