Fundamentals of Coding and Cloud

COURSE OBJECTIVE

- Acquire skills on fundamentals of Python programming, including syntax, data types, and control flow, to build a solid foundation in the language.

COURSE OUTCOMES

- Develop Hands on Experience in object-oriented programming concepts such as classes, objects, inheritance, and polymorphism to write modular and reusable code.
- Develop problem-solving skills by working with data structures like arrays, lists, dictionaries, and implementing searching and sorting algorithms.
- Develop hands on Experience to handle file input/output operations, perform string manipulation, and apply regular expressions for efficient data processing.
- Develop a command over Python programming language, including its syntax, data types, and control flow, to develop robust and efficient code.
- Apply object-oriented programming principles, such as encapsulation, inheritance, and polymorphism, to design and implement modular and reusable code.
- Develop skills in manipulating data structures like arrays, lists, and dictionaries, and implement common algorithms such as searching and sorting for efficient data processing.

COURSE CURRICULUM

UNIT - I	Fundamentals of Python
Introduction to Python: history, features, and Applications-Setting up the development environment for Python-Basic syntax **Variables and data types:** numbers-strings-lists-tuples-dictionaries **Operators:** arithmetic-assignment-comparison-logical-bitwise operators **Control flow:** conditional statements (if, Elif, else) and loops (for and while) **Functions:** defining functions, function parameters, return values, and recursion **File handling:** reading from and writing to files Basic input and output operations.	

UNIT - II	**Object-Oriented Programming (OOP)**

Introduction to OOP: principles- concepts-benefits.

Classes and objects: defining classes-creating objects-accessing attributes and methods.

Encapsulation: data hiding and access modifiers (public, private, protected).

Inheritance: creating derived classes- method overriding-super keywords.

Polymorphism: method overloading-method overriding-abstract classes.

Advanced OOP concepts: interfaces, composition, and static methods.

Challenges: -

1- Implement a class called BankAccount that represents a bank account. The class should have private attributes for account number, account holder name, and account balance. Include methods to deposit money, withdraw money, and display the account balance. Ensure that the account balance cannot be accessed directly from outside the class. Write a program to create an instance of the BankAccount class and test the deposit and withdrawal functionality.

2- Implement a class called Player that represents a cricket player. The Player class should have a method called play() which prints "The player is playing cricket. Derive two classes, Batsman and Bowler, from the Player class. Override the play() method in each derived class to print "The batsman is batting" and "The bowler is bowling", respectively. Write a program to create objects of both the Batsman and Bowler classes and call the play() method for each object.

UNIT - III	**Data Structures and Manipulation**

Arrays and lists: indexing-slicing-common list operations.

String manipulation: string methods-formatting-regular expressions.

Dictionaries and sets: manipulating dictionary -set objects.

Stack and queue data structures.

Searching algorithms: linear search-binary search.

Sorting algorithms: bubble sort- insertion sort- selection sort.

Challenges: -

1-Write a function called linear_search_product that takes the list of products and a target product name as input. The function should perform a linear search to find the target product in the list and return a list of indices of all occurrences of the product if found, or an empty list if the product is not found.

2- Implement a function called sort_students that takes a list of student objects as input and sorts the list based on their CGPA (Cumulative Grade Point Average) in descending order. Each student object has the following attributes: name (string), roll_number (string), and cgpa (float). Test the function with different input lists of students.

REFERENCE	
1	Python Programming: - https://developers.google.com/edu/python
2	Python OOPs: - https://realpython.com/python3-object-oriented-programming/
3	Competitive Programming: - https://www.codewars.com/kata/python
4	Aptitude: - https://www.indiabix.com/aptitude/questions-and-answers/

SOFTWARE REQUIREMENT
● Google Colab / Anaconda Navigator

HARDWARE REQUIREMENT
● Desktop / Laptop with Core i3 or higher Processor, Windows 10 OS / 4 GB of RAM
● Minimum 5 MBPS download speed of internet connection