SET 1

Day 1: Basics of Python

1. **Greeting App**: Ask the user for their name and display a personalized greeting.
2. **Age Calculator**: Calculate the user's age based on their birth year.
3. **BMI Calculator**: Compute Body Mass Index given weight and height.
4. **Tip Calculator**: Split the bill and calculate tips based on total bill and tip percentage.
5. **Number Reverser**: Reverse a given number (e.g., input 123, output 321).

Day 2: Control Flow and Loops

6. **Leap Year Checker**: Check whether a given year is a leap year.
7. **Number Guessing Game**: Let the user guess a number between 1 and 100.
8. **Multiplication Table Generator**: Display the multiplication table for any given number.
9. **Palindrome Checker**: Check if a string or number is a palindrome.
10. **Count Even and Odd Numbers**: Count how many even and odd numbers exist in a given range.

Day 3: Functions and Modules

11. **Password Strength Checker**: Validate if a password meets security criteria (length, symbols, etc.).
12. **Random Name Picker**: Use the random module to pick a name from a list.
13. **Prime Number Finder**: Create a function to find all prime numbers within a range.
14. **Unit Converter**: Convert units like kilometers to miles, or pounds to kilograms.
15. **Dice Roller**: Simulate rolling a pair of dice and display the result.

Day 4: Data Structures

16. **Shopping Cart**: Allow users to add, remove, and view items in a shopping list (using lists).
17. **Word Frequency Counter**: Count the frequency of words in a sentence (using dictionaries).
18. **Matrix Addition**: Perform addition of two matrices (using nested lists).
19. **Phonebook Application**: Use a dictionary to add, delete, and search contacts.
20. **Sorting Tool**: Sort a list of numbers in ascending or descending order.

Day 5: File Handling and Error Handling

21. **Quiz App**: Store quiz questions and answers in a file and retrieve them for a quiz game.
22. **Log Writer**: Append log entries to a file whenever an error occurs.
23. **Character Counter**: Count the number of characters, words, and lines in a file.
24. **File Copier**: Copy the contents of one file to another.
25. **JSON Data Manager**: Read and write data to a JSON file.

Day 6: Object-Oriented Programming

26. **Library Book System**: Create a class to represent books with properties like title, author, and availability.
27. **Vehicle Inheritance**: Create a Vehicle base class and derived classes for Car and Bike.
28. **Employee Payroll System**: Create an employee class to manage salaries and bonuses.
29. **Attendance Manager**: Track attendance for a group of students using classes and methods.
30. **Inventory Manager**: Manage inventory items with properties like item name, quantity, and price.

SET 2

Day 1: Basics of Python

1. **Hello, World! App**: Display a welcome message on the screen.

2. **Basic Calculator**: Perform addition, subtraction, multiplication, and division.
3. **Temperature Converter**: Convert Celsius to Fahrenheit and vice versa.
4. **Area of a Circle**: Calculate the area of a circle given its radius.
5. **Simple Interest Calculator**: Compute the simple interest for given principal, rate, and time.

Day 2: Control Flow and Loops

6. **Even or Odd Checker**: Determine if a number is even or odd.
7. **Grade Calculator**: Assign grades based on marks input by the user.
8. **Factorial Calculator**: Calculate the factorial of a number using loops.
9. **Fibonacci Series Generator**: Generate the first N terms of the Fibonacci sequence.
10. **Sum of Numbers in a Range**: Find the sum of all numbers in a user-defined range.

Day 3: Functions and Modules

11. **Prime Number Checker**: Create a function to check if a number is prime.
12. **Math Solver**: Use the math module to calculate square roots, trigonometric values, etc.
13. **Fibonacci Function**: Write a function to generate Fibonacci numbers.
14. **Custom Greetings**: Create a function to display personalized greeting messages.
15. **Random Password Generator**: Use the random module to create a secure password.

Day 4: Data Structures

16. **List Manager**: Create, modify, and display a list of items.
17. **Dictionary-based Address Book**: Manage an address book using a dictionary.
18. **Set Operations**: Perform union, intersection, and difference of two sets.

19. **Student Grade Tracker**: Use a dictionary to store and retrieve student grades.
20. **Common Elements in Two Lists**: Find and display common elements between two lists.

Day 5: File Handling and Error Handling

21. **File Reader**: Read the contents of a text file and display them.
22. **Diary App**: Write user inputs into a file to maintain a diary.
23. **Error Handling App**: Handle errors while dividing two numbers.
24. **User Feedback Recorder**: Save user feedback into a file and read it back.
25. **CSV File Processor**: Read and display data from a CSV file.

Day 6: Object-Oriented Programming

26. **Bank Account Manager**: Create a class to manage deposits and withdrawals.
27. **Student Information System**: Implement a class for students with details like name, age, and ID.
28. **Basic Calculator Using OOP**: Develop a calculator with methods for different operations.
29. **Shape Area Calculator**: Create a class hierarchy for shapes and calculate areas of circles and rectangles.
30. **Library Management System**: Use classes to manage books and users.