Assignment

1. Write a Python program to check whether a given number is prime or not.

   Input: Take an integer n as input from user.

   Output: Prints "Prime" if the number is prime, otherwise "Not Prime".

   - **Sample Input:** 7
   - **Sample Output:** Prime

2. Write a Python program that prints the multiplication table of a given number.

   Input: Take an integer n as input from user.

   Output: Prints the multiplication table of n from 1 to 10.

   - **Sample Input:** 3
   - **Sample Output:** 3 x 1 = 3
     3 x 2 = 6

3. Write a Python program that counts the number of digits in a given number.

   Input: Take a string s as input from user.

   Output: Prints the number of digits.

   - **Sample Input:** ab12345
   - **Sample Output:** Total digits are : 5

4. Write a Python function to check if a given string is a palindrome. A string is a palindrome if it reads the same forward and backward.

   Input: Take a string as input from a user.

   Output: Return **True** if the string is a palindrome, otherwise **False**.

   - **Sample Input:** "madam"
   - **Sample Output:** True

5. Write a Python function that counts the number of vowels in a given string.
    - **Sample Input:** "hello"
    - **Sample Output:** Number of vowels : 2
6. Write a Python function to find the greatest common divisor (GCD) of two numbers a and b.
    - **Sample Input:** 12,18
    - **Sample Output:** GCD of 12 and 18 is : 6

7. Write a Python function to find the second largest number in a list.
   Input: Take a list as input from user.
   Output: The second largest number in the list.
    - **Sample Input:** [1, 22, 35, -10, 7]
    - **Sample Output:** 22
8. **Problem Statement:** Create a class called Book that represents a book. The class should have the following attributes:

- title (string)
- author (string)
- pages (integer)

Implement the following methods:

- __init__: Initializes the book with title, author, and pages.
- get_description: Returns a string description of the book in the format "Title by Author, Pages: X".
- **Sample Input:** book1 = Book("1984", "George Orwell", 328)
    - print(book1.get_description())
- **Sample Output:** 1984 by George Orwell, Pages: 328

9. **Problem Statement:** Create a base class called Animal with the following methods:

- __init__: Initializes the animal with a name and species.
- make_sound: Returns a generic sound.

Create a derived class called Dog that overrides make_sound to return "Woof!". Also, implement a method fetch that returns a string saying the dog is fetching a ball.

- **Sample Input:** dog = Dog("Buddy", "Golden Retriever")
  print(dog.make_sound())
  print(dog.fetch())
- **Sample Output:** Woof!
  Buddy is fetching the ball.

10. **Problem Statement:** Create a class called BankAccount that has a private attribute balance. Implement the following methods:

- __init__: Initializes the account with a starting balance.
- deposit: Adds an amount to the balance.
- withdraw: Subtracts an amount from the balance if sufficient funds are available.
- get_balance: Returns the current balance.

Sample Input: account = BankAccount(1000)
account.deposit(500)
print(account.get_balance())
account.withdraw(300)
print(account.get_balance())
account.withdraw(1500)

Sample Output: 1500
1200
Insufficient funds!