Assignment: Python Fundamentals

Note: For the assignment, you need to solve the given problem logically and also build a Python project according to standard coding practices. Once the project and assignment is complete, push the code to GitHub. After pushing, copy the repository link and send via email saishna.budhathoki@gmail.com by **Saturday**. Make sure the GitHub repository is public or shared as required, and include the link in the email along with any relevant details. Ensure everything is submitted before the deadline.

Section 1: Conditional Statement

1. **Write a program that asks the user to enter a number and checks whether the number is even or odd.**
2. **Write a program that asks the user to enter a number and checks whether it is positive, negative, or zero.**
3. **Write a program that asks the user to enter their age and checks if they are eligible to vote (18 years or older).**
4. **Write a program that takes the lengths of three sides of a triangle as input and determines if the triangle is:**
    - **Equilateral (all sides are equal)**
    - **Isosceles (two sides are equal)**
    - **Scalene (all sides are different)**
5. **Write a program that asks the user to enter a password and checks if it matches a predefined password (e.g., "Python123").**

Section 2: For Loops

6. **Multiplication Table: Write a Python program to print the multiplication table of a given number using a for loop.**
   Example:
   - Input: 5
   - Output: 5, 10, 15, ..., 50

7. **Count Vowels: Write a program to count and display the number of vowels in a given string.**

8. **Sum of Even Numbers Create a program to find the sum of all even numbers in a list using a for loop.**

   Example:

   - Input: [2, 5, 8, 3]
   - Output: 10

9. **Create a program to find the largest and smallest numbers in a list using a for loop.**

   Example:

   - Input = [10, 20, 5, 8], Output = Largest = 20, Smallest = 5

10. **Write a program to calculate the sum of digits of a given number using a for loop.**

    Example: Input = 1234, Output = 10

11. **Write a program to reverse a given string using a for loop.**

    Example: Input = "Python", Output = "nohtyP"

Section 3: Functions

1. **Factorial Function: Write a function factorial(n) that returns the factorial of a number using a for loop.**

2. **Prime Check Function: Create a function is_prime(n) to check whether a number is prime.**

   Example:

   - Input: 7
   - Output: True

3. **Find Maximum in List:Write a function find_max(lst) to find and return the maximum value in a list.**

   Example:

   - Input: [1, 3, 7, 0]
   - Output: 7

4. **Write a function palindrome_check(string) that returns True if the given string is a palindrome, otherwise False.**

   Example: Input = "radar", Output = True

5. **Write a function sum_of_squares(n) that returns the sum of squares of the first n natural numbers.**

6. **Write a function sum_of_squares(n) that returns the sum of squares of the first n natural numbers.**

   Example: Input = 3, Output = 1^2 + 2^2 + 3^2 = 14

Section 4: Classes

1. **Rectangle Class:Define a class Rectangle with:**
   - **Two attributes: length and width.**
   - **A method area() to calculate the area.**
   - **A method perimeter() to calculate the perimeter.**

2. **Student Class: Create a class Student with:**
   - **Attributes: name, roll_number, and marks.**
   - **A method display_details() to print the student's details.**
   - **A method is_passed() that returns True if marks ≥ 40, else False.**

3. **Define a class Circle with:**
   - **Attribute: radius.**
   - **Methods: area() to calculate the area, circumference() to calculate the circumference.**

4. **Create a class BankAccount with:**
   - **Attributes: account_number, account_holder, balance.**
   - **Methods:**
     - **deposit(amount) to add money to the account.**

    - withdraw(amount) to subtract money (if sufficient balance is available).
    - display_balance() to print the current balance.

5. A class Book with:
    - Attributes: title, author, price.
    - Methods:
        - apply_discount(discount) to reduce the price by a given percentage.
        - display_details() to print the book details.

6. Create a class Calculator with:
    - A method add(a, b) to add two numbers.
    - A method subtract(a, b) to subtract two numbers.
    - A method multiply(a, b) to multiply two numbers.
    - A method divide(a, b) to divide two numbers (handle division by zero gracefully).

7. Write a class Person with:
    - Attributes: name, age, gender.
    - Method introduce() to print a message like: "Hello, my name is [name], I am [age] years old, and I am a [gender]."
    - Create a subclass Employee that adds:
        - Attribute: job_title.
        - Override introduce() to include the job title.

Section 4: File Handling

1. **File Creation and Writing: Write a Python script to:**
    - **Create a file named data.txt.**
    - **Write the numbers from 1 to 10 (each on a new line) to the file.**
    - **Read the file and print its contents.**

2. **Word Count in File: Write a program that reads a file named sample.txt, counts the number of words in it, and prints the result.**

Section 5: Error Handling

1. **Division with Error Handling: Write a program that takes two numbers as input and divides the first number by the second.**
    - Use try and except to handle the division by zero error.

2. **Integer Input Validation: Create a program that asks the user to input an integer.**
    - **Use try, except, and else to handle invalid input.**

    Example:
    - Input: abc
    - Output: "Invalid input, please enter an integer."

3. **File Not Found Error Handling: Write a script that:**

- **Opens a file named unknown.txt.**
- **Handles the FileNotFoundError gracefully and prints an appropriate message.**