# Topics to Revise

* - Python Programming: List comprehensions, OOP concepts, control structures, data structures, string operations, error handling.
* - Agile and Software Development Methodologies: Scrum practices, Agile vs. Waterfall, DevOps principles.
* - Algorithms and Logic: Graph traversal, truth tables, recursion.
* - Python Concepts: Functional programming, defining and using functions, Python class essentials.
* - Problem Solving: Code tracing, debugging, logic evaluation.
* - Additional Topics: Usability and user interface (UI) design principles.

# Revision Worksheet: Problem Solving and Python Concepts

## Section 1: Python Programming

1. 1. Write a list comprehension that generates a list of squares for numbers from 1 to 10.
2. 2. What does the `\_\_init\_\_` method do in Python classes? Write an example of a class with an `\_\_init\_\_` method.
3. 3. Write a Python script that prints all even numbers between 1 and 20 using a `for` loop.
4. 4. Create a dictionary that maps days of the week to their abbreviations (e.g., Monday -> Mon). Write code to add a new key-value pair to this dictionary.
5. 5. Given the string `text = "Hello, Python!"`, write code to:  
   - Extract the word "Python".  
   - Replace "Python" with "World".
6. 6. Write a `try-except` block to handle a division by zero error in Python.

## Section 2: Agile and Software Development Methodologies

1. 1. List three key practices of Scrum methodology.
2. 2. Compare Agile and Waterfall development methodologies in two sentences.
3. 3. Explain Continuous Integration and Continuous Delivery (CI/CD) in your own words.

## Section 3: Algorithms and Logic

1. 1. Name and describe the Breadth-First Search (BFS) algorithm.
2. 2. Create a truth table for an AND gate with two inputs.
3. 3. Write a Python function to calculate the factorial of a number using recursion.

## Section 4: Python Concepts

1. 1. Write a Python lambda function to add two numbers.
2. 2. Define a Python function that accepts a list of numbers and returns their sum.
3. 3. Write a class `Car` with attributes `make`, `model`, and `year`. Include a method to display the car's details.

## Section 5: Problem Solving

1. 1. What is the output of this code snippet? Explain your answer.  
   ```python  
   lst = [x \* 2 for x in range(3)]  
   print(lst)  
   ```
2. 2. Fix the error in the following code:  
   ```python  
   def greet():  
   print("Hello, World!")  
   ```

## Section 6: Additional Topics

1. 1. Describe one principle of good user interface design.