Computer Science and Programming 24-25 (Johnson)

Name:____

Period:________

Unit 8 Lab 1: Working with Loops

Objective:

Students will demonstrate an understanding of loops in python by implementing a loop in place of inefficient code design and by debugging and correcting erroneous code.

Part A: Using a for Loop to Improve Efficiency

The program below (found in loopLabPartA.py) prints out the squares of the first 25 numbers (1 through 25), but it uses repeated code. Refactor the code to make it more efficient by using a for loop.

```
# Initial repetitive code:

print(1 ** 2)

print(2 ** 2)

…

print(24 ** 2)

print(25 ** 2)
```

Instructions:

1. Identify the repeated pattern in the code.

   ____

2. Explain how you plan to refactor (replace) it.

   ____

3. Refactor the code in loopLabPartA.py based on your plan above.
4. Test the program and push code into GitHub.

Part B: Fixing a while Loop Bug

The following while loop (found in loopLabPartB.py) is supposed to simulate a parking meter. However, there are two errors, and the loop doesn't work as intended. Identify and correct the error.

```
parking_time = 10  # 10 minutes remaining on the meter

coin_inserted = False

print("Parking meter started. Time remaining:")

while parking_time >= 0:

    print(f"{parking_time} minutes left.")

    if parking_time % 2 == 0:

        parking_time += 1

    if coin_inserted:

        parking_time += 5  # Add 5 more minutes if a coin is inserted

        coin_inserted = False  # Reset after coin is used

print("Parking time expired. Please insert more coins.")
```

Instructions:

1. Identify the first error that causes the loop to run infinitely. Explain how you intend to fix it.

   ____

   ____

2. Identify the second error that causes the loop to run an extra time. Explain how you intend to fix it.

   ____

   ____

3. Modify the code based on your plans above.
4. Test the program and push code into GitHub.

Extension (Optional 4 pts extra credit):

- Modify the program in Part A to accept user input for the range of numbers to square.

  (2 pts)

- Modify the program In Part B to accept user input for adding a coin when time has run out.

  (2pts)

Computer Science and Programming 24-25 (Johnson)

Rubric for Loops Lab (Total: 12 Points)

Criteria	Score	4 Points	3 Points	2 Points	1 Point	0 Points
Part A Task Completion		All tasks completed	3 tasks completed	2 tasks completed	1 Task completed	No tasks completed.
Part B Task Completion		All tasks completed	3 tasks completed	2 tasks completed	1 Task completed	No tasks completed.
Code Functionality		Code runs without errors and as expected.	Code runs with minor errors but achieves the goal.	Code runs but has significant errors affecting functionality.	Code does not run.	No code submitted.
Extra Credit		Both parts implemented and functional		One part implemented and functional		No attempt
Total Sore						