# ASSIGNMENT 7.3 2303A52490

## Task 1: Fixing Syntax Errors

**Original buggy code (missing colon):**

```
def add(a, b)
    return a + b
```

**AI Prompt used:** "Fix this Python syntax error in the function definition."

**Terminal execution of buggy code:**

```
$ python3 buggy_syntax.py
  File "buggy_syntax.py", line 1
    def add(a, b)
                ^
SyntaxError: invalid syntax
```

**AI Corrected code:**

```
def add(a, b):
    return a + b
```

**Terminal execution of corrected code:**

```
$ python3 fixed_syntax.py
5
```

**AI Explanation:** The colon after the function parameters is missing, which is required for Python function definitions to indicate the start of the function body. Added the colon to fix the SyntaxError.

## Task 2: Debugging Logic Errors in Loops

**Original buggy code (infinite loop):**

```
count = 10
while count > 0:
    print(count)
    # count -= 1  # forgotten decrement
```

**AI Prompt used:** "Debug this infinite loop in Python."

**Terminal execution (hangs, Ctrl+C to stop):**

```
$ python3 buggy_loop.py
10
10
10
^CTraceback (most recent call last):
  File "buggy_loop.py", line 3, in <module>
    print(count)
KeyboardInterrupt
```

**AI Corrected code:**

```
count = 10
while count > 0:
    print(count)
    count -= 1
```

**Terminal execution of corrected code:**

```
$ python3 fixed_loop.py
10
9
8
7
6
5
4
3
2
1
```

**AI Explanation:** The loop lacks a decrement on 'count', causing infinite iteration since the condition never becomes false. Added 'count -= 1' inside the loop to fix the logic error.

## Task 3: Handling Runtime Errors - Division by Zero

**Original buggy code:**

```
def divide(x, y):
    return x / y

print(divide(10, 0))
```

**AI Prompt used:** "Fix division by zero runtime error in this function."

**Terminal execution of buggy code:**

```
$ python3 buggy_divide.py
Traceback (most recent call last):
  File "buggy_divide.py", line 4, in <module>
    print(divide(10, 0))
            ~~~~~~~~~~~~
  File "buggy_divide.py", line 2, in divide
    return x / y
           ~~^~~
ZeroDivisionError: division by zero
```

**AI Corrected code:**

```
def divide(x, y):
    try:
        return x / y
    except ZeroDivisionError:
        return "Cannot divide by zero"

print(divide(10, 0))
```

**Terminal execution of corrected code:**

```
$ python3 fixed_divide.py
Cannot divide by zero
```

**AI Explanation:** Added try-except block to catch ZeroDivisionError specifically. Returns a safe message instead of crashing when denominator is zero.

## Task 4: Debugging Class Definition Errors

**Original buggy code (missing self):**

```
class Student:
    def __init__(name):
        self.name = name
```

**AI Prompt used:** "Correct this Python class constructor error."

**Terminal execution of buggy code:**

```
$ python3 buggy_class.py
Traceback (most recent call last):
  File "buggy_class.py", line 5, in <module>
    s = Student("Alice")
TypeError: Student.__init__() missing 1 required positional argument: 'name'
```

**AI Corrected code:**

```
class Student:
    def __init__(self, name):
        self.name = name


s = Student("Alice")
print(s.name)
```

**Terminal execution of corrected code:**

```
$ python3 fixed_class.py
Alice
```

**AI Explanation:** The **init** method requires 'self' as the first parameter to refer to the instance. Added 'self' to fix the TypeError.

## Task 5: Resolving Index Errors in Lists

**Original buggy code:**

```
my_list = [1, 2, 3]
print(my_list[^1_5])
```

**AI Prompt used:** "Fix index out of range error in this list access."

**Terminal execution of buggy code:**

```
$ python3 buggy_index.py
Traceback (most recent call last):
  File "buggy_index.py", line 2, in <module>
    print(my_list[^1_5])
IndexError: list index out of range
```

**AI Corrected code:**

```
my_list = [1, 2, 3]
if len(my_list) > 5:
    print(my_list[^1_5])
else:
    print("Index out of range")
```

**Terminal execution of corrected code:**

```
$ python3 fixed_index.py
Index out of range
```

**AI Explanation:** Checked list length before accessing index to prevent IndexError. Used conditional bounds checking for safe access.