Lab Assignment - 7

Error Debugging with AI: Systematic approaches to finding and fixing bugs

TASK-1 – Runtime Error Due to Invalid Input Type

Error code:

```
num = input("Enter a number: ")
result = num + 10
print(result)
```

4.7s Python

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[1], line 2
      1 num = input("Enter a number: ")
----> 2 result = num + 10
      3 print(result)

TypeError: can only concatenate str (not "int") to str
```

AI Generated code:

```
# TypeError: the original code attempted to add an int (10) to a str (input), which is not allowed.
# Fix: convert the input string to a numeric type (int or float) before performing arithmetic.
num = input("Enter a number: ")  # e.g. '10'
try:
    value = int(num)
except ValueError:
    # ValueError: input couldn't be parsed as int; try float (e.g. '3.14')
    try:
        value = float(num)
    except ValueError:
        raise ValueError(f"Invalid numeric input: {num!r}")

result = value + 10
print(result)
```

✓ 3.7s Python

```
20
```

TASK-2 – Incorrect Function Return Value

Error(Buggy) code:

```
def square(n):
result = n * n
```

0.0s

```
  Cell In[4], line 2
    result = n * n
    ^
IndentationError: expected an indented block after function definition on line 1
```

AI Generated code:

```
def square(n):
    return n * n

# Example using existing `value` variable from the notebook
square_value = square(value)
print(square_value)
```

✓ 0.0s

TASK-3 – IndexError in List Traversal

Error Code:

```
numbers = [10, 20, 30]
for i in range(0, len(numbers)+1):
    print(numbers[i],end=" ")
```

0.0s

```
10 20 30

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[6], line 3
      1 numbers = [10, 20, 30]
      2 for i in range(0, len(numbers)+1):
----> 3     print(numbers[i],end=" ")

IndexError: list index out of range
```

AI Generated code:

```
# BUG FIX: original used range(0, len(numbers)+1) which goes out of bounds when i == len(numbers)
# Fix: use range(len(numbers)) so indices go from 0 to len(numbers)-1 (no IndexError)
for i in range(len(numbers)):
    print(numbers[i],end=" ")

# Alternative (recommended): iterate directly over elements to avoid index handling entirely
# for value in numbers:
#     print(value)
```

✓ 0.0s Python

```
10 20 30
```

TASK-4 – Uninitialized Variable Usage

Error Code:

```
if True:
    pass
print(total)
```

⊗ 0.0s

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 3
      1 if True:
      2     pass
----> 3 print(total)

NameError: name 'total' is not defined
```

AI Generated Code:

```
# BUG: original code used `total` before it was defined, causing NameError ("name 'total' is not defined")
# FIX: initialize `total` before use. Here we set it to the sum of the existing `numbers` list.
total = sum(numbers)

if True:
    pass

print(total)  # prints initialized total (e.g., 60)
```

✓ 0.0s Python

```
60
```

TASK-5 – Logical Error in Student Grading System

Buggy code:

```
marks = 85
if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "C"
else:
    grade = "B"
print(grade)
```

0.0s

```
C
```

The grading logic is wrong because students who score between 80 and 89 get grade C instead of B. Also, students scoring below 80 are incorrectly given grade B.

AI Corrected Logic code:

```
# BUG FIX: The original code assigned "C" for marks >= 80, and "B" for marks < 80.
# Typically, grading logic is: A (>=90), B (>=80), C (>=70), etc.
# The bug was that "C" was assigned for marks >= 80, which should be "B".
# Fix: Assign "B" for marks >= 80, and "C" for marks >= 70.

if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "B"
elif marks >= 70:
    grade = "C"
else:
    grade = "D"

print(grade)
```

0.0s

```
B
```