CSC500 – Week 2 Assignment

Student: Cervonte Allen
Topic: Short-circuit evaluation with AND / OR

Extended Description (Simple Words)

Python can stop early when checking combined conditions. With AND, if the left side is False, it skips the right side. With OR, if the left side is True, it skips the right side. This saves time and can prevent errors (like dividing by zero).

Part 1: Short-circuit Basics

Pseudocode

```
PRINT 'AND stops on first False'
SET result1 = left_false() AND right_side()   # right_side() does not run
PRINT result1

PRINT 'OR stops on first True'
SET result2 = left_true() OR right_side()     # right_side() does not run
PRINT result2

PRINT 'OR when left is False -> must check right'
SET result3 = left_false() OR right_side()    # right_side() runs
PRINT result3
```

Python Source Code

```
# Week 2 - Part 1: Short-circuit basics with AND / OR
def left_true():
    print("left_true() ran")
    return True

def left_false():
    print("left_false() ran")
    return False

def right_side():
    print("right_side() ran")
    return True

def main():
    print("Case 1: AND stops on first False")
    result_and = left_false() and right_side()
    print("Result AND:", result_and)
    print()

    print("Case 2: OR stops on first True")
    result_or = left_true() or right_side()
    print("Result OR:", result_or)
    print()

    print("Case 3: OR when left is False -> must check right")
    result_or2 = left_false() or right_side()
    print("Result OR2:", result_or2)

if __name__ == "__main__":
    main()
```

Sample Run (text)

```
Case 1: AND stops on first False
left_false() ran
Result AND: False

Case 2: OR stops on first True
left_true() ran
Result OR: True

Case 3: OR when left is False -> must check right
left_false() ran
right_side() ran
Result OR2: True
```

Part 2: Guard Patterns (Avoid Errors)

Pseudocode

```
FUNCTION safe_divide(a, b):
    IF b != 0 AND (a / b) is computed THEN RETURN result
    ELSE RETURN None

FUNCTION has_valid_name(user):
    RETURN (user is not None) AND user has a non-empty 'name'

MAIN:
    PRINT results of safe_divide(10,2) and safe_divide(5,0)
    PRINT results of has_valid_name(User('Alex')) and has_valid_name(None)
```

Python Source Code

```
# Week 2 - Part 2: Guard patterns using short-circuiting
def safe_divide(a, b):
    # If b is zero, the right side won't run; return None
    return (b != 0) and (a / b) or None

def has_valid_name(user):
    return (user is not None) and bool(getattr(user, "name", ""))

class User:
    def __init__(self, name):
        self.name = name

def main():
    print("Safe divide examples:")
    print("10 / 2 ->", safe_divide(10, 2))
    print("5 / 0  ->", safe_divide(5, 0))
    print()

    print("Validate user name safely:")
    u1 = User("Alex")
    u2 = None
    print("u1 has name? ->", has_valid_name(u1))
    print("u2 has name? ->", has_valid_name(u2))

if __name__ == "__main__":
    main()
```

Sample Run (text)

```
Safe divide examples:
10 / 2 -> 5.0
5 / 0  -> None
Validate user name safely:
```

```
u1 has name? -> True
u2 has name? -> False
```