

AI-Generated Code Review with Errored Code

Prompt Used:

“Find and fix syntax errors, function name mistakes, and logic problems”

Code:

```
import numpy as np
import pandas as pd
import random

def generate_random_number(min_num, max_num):
    num = random.randint(min_num, max_num)
    print("Random number is: " + num)

def calc_average(num_list):
    total = sum(num_list)
    return total / lenght(num_list)

def check_prime(start, end):
    prime_list = []
    for i in range(start, end):
        if i % 2 == 0:
            prime_list.append(i)
    return prime_list

def load_data(filepath):
    data = pd.read_csv(filepath)
    return data

def main():
    num_list = [10, 20, 30, "forty", 50]
    print("The average is: ", calc_average(num_list))
    print("Prime numbers: ", check_prime(1, 10))

    file_path = "data.csv"
    data = load_data(file_path)
    print("Data loaded: ", data)

    random_num = generate_random_number(1, 100)
    print("Generated Random Number: ", random_num)
```

```
try:
    print("Result of division: ", 10 / 0)
except ZeroDivisionError:
    print("Can't divide by zero")

numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]
print("Numbers divisible by 3 and 5 are: ", numbers)

undefined_function_call()

main()
```

Observations:



1. ChatGPT

I've found a few issues in the code and will fix them one by one.

Issues:

- generate_random_number function:** The `num` variable is an integer, and you are trying to concatenate it with a string, which will cause an error.
 - Solution: Convert the integer to a string before concatenation.
- calc_average function:** The function uses `lenght()` instead of `len()`, which causes a syntax error.
 - Solution: Correct `lenght` to `len()`.
- check_prime function:** The logic for checking primes is incorrect. It only appends even numbers to the list, but prime numbers (except for 2) are not even.
 - Solution: Modify the logic to check for prime numbers.
- generate_random_number return value:** It currently prints the number but doesn't return anything.
 - Solution: Ensure the function returns the random number.
- undefined_function_call() :** This function doesn't exist in the code.
 - Solution: Remove or replace with a valid function.

python

 Copy  Edit

```
import numpy as np
import pandas as pd
import random

def generate_random_number(min_num, max_num):
    num = random.randint(min_num, max_num)
    print("Random number is: " + str(num)) # Convert num to string for concatenation
    return num # Return the random number

def calc_average(num_list):
    total = sum(num_list)
    return total / len(num_list) # Corrected 'lenght' to 'len'

def check_prime(start, end):
    prime_list = []
    for i in range(start, end):
        if i > 1: # Numbers less than 2 are not prime
            for j in range(2, i):
                if i % j == 0:
                    break
            else:
                prime_list.append(i)
    return prime_list
```

```
def load_data(filepath):
    data = pd.read_csv(filepath)
    return data

def main():
    num_list = [10, 20, 30, 40, 50] # Fixed string "forty" to an integer
    print("The average is: ", calc_average(num_list))
    print("Prime numbers: ", check_prime(1, 10))

    file_path = "data.csv"
    data = load_data(file_path)
    print("Data loaded: ", data)

    random_num = generate_random_number(1, 100)
    print("Generated Random Number: ", random_num)

    try:
        print("Result of division: ", 10 / 0)
    except ZeroDivisionError:
        print("Can't divide by zero")

    numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]
    print("Numbers divisible by 3 and 5 are: ", numbers)

    # Removed undefined function call

main()
```

Key Changes:

1. Concatenation in `generate_random_number`: `str(num)` is used to convert the number to a string.
2. `lenght` typo fixed to `len()` in `calc_average`.
3. Prime checking logic in `check_prime`: The algorithm was updated to check for prime numbers properly.
4. Fixed list `num_list`: Replaced the string "forty" with an integer to avoid issues during averaging.
5. Removed `undefined_function_call()` as it is not defined.

This should work correctly now! Let me know if you have further questions or issues.



2. Perplexity.ai

✳ Answer

Okay, I will review the code you provided, identify syntax errors, function name mistakes, logic problems, and provide a corrected version with explanations.

python



```
import numpy as np
import pandas as pd
import random

def generate_random_number(min_num, max_num):
    num = random.randint(min_num, max_num)
    print("Random number is: " + str(num)) # Corrected: Convert num to
string for concatenation

def calc_average(num_list):
    total = sum(num_list)
    return total / len(num_list) # Corrected: Use len() to get the
length of the list

def check_prime(start, end):
    prime_list = []
    for i in range(start, end + 1): # Corrected: Iterate through the
specified range (inclusive of end)
        if i > 1: # ensures numbers 0 and 1 are not checked
            for j in range(2, int(i**0.5) + 1): # only check divisibility
up to the square root of the number
                if (i % j) == 0:
                    break
            else:
                prime_list.append(i)
    return prime_list
```

```
def load_data(filepath):
    try: # Added: Handle FileNotFoundError
        data = pd.read_csv(filepath)
        return data
    except FileNotFoundError:
        print(f"Error: File not found at {filepath}")
        return None

def main():
    num_list = [10, 20, 30, 40, 50] # Corrected: "forty" changed to 40
    to allow sum
    print("The average is: ", calc_average(num_list))
    print("Prime numbers: ", check_prime(1, 100)) # Corrected: set the
    end to 100 to see prime numbers

    file_path = "data.csv"
    data = load_data(file_path)
    if data is not None: # Added: Check if data loading was successful
        print("Data loaded: ", data)

    random_num = generate_random_number(1, 100)

    try:
        print("Result of division: ", 10 / 0)
    except ZeroDivisionError:
        print("Can't divide by zero")

    numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]
    print("Numbers divisible by 3 and 5 are: ", numbers)

    # Corrected: Removed undefined function call to avoid NameError
    # undefined_function_call()

main()
```

Explanation of Corrections:

1. `generate_random_number` function:

- **Type Error:** The original code had a `TypeError` because it tried to concatenate a string with an integer directly. I fixed this by converting the integer `num` to a string using `str(num)` before concatenating it with the rest of the string in the `print` statement.

2. `calc_average` function:

- **Name Error:** The original code had a `NameError` because it used `lenght()` instead of `len()` to get the length of the list. I fixed this by changing `lenght(num_list)` to `len(num_list)`.
- **Type Error:** The original list contained the string `"forty"` which cannot be summed with integers. I replaced this with `40` to avoid the `TypeError`.

3. `check_prime` function:

- **Logic Error:** The original code only checked if numbers were divisible by 2, which is not sufficient to determine if a number is prime. I corrected this by iterating through the range of numbers and checking for divisibility from 2 up to the square root of each number. If a number is divisible by any number in this range, it's not prime.
- **Range Error:** The function did not include the end number, meaning that the number 9 would never be checked. To fix this, I changed the end range to `end + 1`.

4. `load_data` function:

- **File Handling:** Added a `try...except` block to handle the case where the file specified by `filepath` does not exist. This prevents the program from crashing and prints an error message instead. The function now returns `None` if the file is not found, which is then checked in `main()`

5. `main` function:

- **Data Validation:** Added a check to ensure that the data was loaded successfully before attempting to print it. This prevents a `NameError` if the `load_data` function returns `None`.
- **Error Handling:** The `ZeroDivisionError` exception handling is already in place and works correctly.
- **Function Call Error:** Removed the call to `undefined_function_call()` because it was causing a `NameError`.
- **Prime number generation:** The upper range for the prime number generation was too small, and did not show any prime numbers. I updated this from 10 to 100 to demonstrate the functionality of the prime number function.

These corrections should resolve the syntax errors, function name mistakes, and logic problems in your code.

Fixed Code Result:

```
Name, Age
Alice, 30
Bob, 25
Charlie, 35
```

For CSV of details, the output shown is:

```
The average is: 30.0
Prime numbers: [2, 3, 5, 7]
Random number is: 45
Generated Random Number: 45
Can't divide by zero
Numbers divisible by 3 and 5 are: [0, 15, 30, 45, 60, 75, 90]
Data loaded:
  Name  Age
0  Alice  30
1   Bob   25
2 Charlie 35
```


Observations:

The fixed code has no errors, and hence completes the required task and calculations are done correctly. All functions in the fixed code work as expected, whereas the broken code has logic errors and unhandled cases. By fixing the broken code, it can be seen that the application runs smoothly and produces the intended results, while the broken code halts at various points due to unhandled errors.