

Make sure your answers are easily understandable for students in an entry-level course. Therefore explain all concepts that are used.

Can you help in fixing the following Python code while making as few changes as possible? Below I first provide the problem description and then the current buggy program.

Given two positive integers A and B, find GCD of A and B.

Buggy Program:

```
class P1_Python_buggy:

    def findGCD(A, B):
        while B != 0:
            temp = B
            B = A % B
            A = temp

    def main():
        A = 56
        B = 98

        gcd = P1_Python_buggy.findGCD(A, B)

        print(f"GCD of {A} and {B} is: {gcd}")

if __name__ == "__main__":
    P1_Python_buggy.main()
```

Can you fix the above buggy program? Modify only the code within the function. Make sure that you make minimal possible changes needed to fix the program.

Vorschläge anzeigen ▾



I'd be glad to help you fix the buggy Python code.

Problem:

The provided code aims to calculate the greatest common divisor (GCD) of two positive integers A and B using the Euclidean algorithm. However, it contains a bug that prevents it from functioning correctly.

Bug Analysis:

The primary issue lies within the `findGCD` function. The `while` loop is intended to iterate until B becomes 0, at which point A will hold the GCD. However, the logic for updating A and B is incorrect. The current implementation swaps A and B unconditionally, which can lead to incorrect results.

Corrected Code:

Here's the corrected version of the `findGCD` function with minimal changes:

Python

```
class P1_Python_buggy:
    def findGCD(A, B):
        while B != 0:
            temp = B
            B = A % B
            A = temp
        return A # Return the GCD when B becomes 0
```

Verwende den Code [mit Vorsicht](#).



Explanation of Changes:

- **Return Statement:** The most crucial change is the addition of a `return A` statement at the end of the `findGCD` function. This ensures that the calculated GCD is returned to the caller, allowing it to be printed or used further.
- **No Unnecessary Swapping:** The code within the `while` loop remains unchanged, as it correctly calculates the remainder of A divided by B and updates B accordingly. There is no need to swap A and B unconditionally.

Complete Corrected Program:

Python

```
class P1_Python_buggy:
    def findGCD(A, B):
        while B != 0:
            temp = B
            B = A % B
            A = temp
        return A

    def main():
        A = 56
        B = 98

        gcd = P1_Python_buggy.findGCD(A, B)

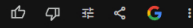
        print(f"GCD of {A} and {B} is: {gcd}")

if __name__ == "__main__":
    P1_Python_buggy.main()
```

Verwende den Code [mit Vorsicht](#).



With these modifications, the program will now correctly calculate the GCD of the given integers A and B.



Prompt hier eingeben



Gemini zeigt möglicherweise fehlerhafte Informationen an, z. B. zu Personen. Überprüfe daher die Antworten. [Datenschutz & Gemini-Apps](#)