

Question: You have been tasked with reviewing and updating this program. What steps would you take to identify any issues in the code?

Your Response: Read through each line, checking for common errors such as incorrect or missing variables, misspellings, arithmetic mistakes, and missing inputs or outputs.

Question: What actions could the developer have taken to minimize the number of errors?

Your Response: They could program in pairs so that the other person is proof reading the code while the author is writing. Additionally, they should take their time and be aware of what they are writing, checking for spelling as they go.

Debugging Exercise:

```
// This pseudocode is intended to determine whether students have
// passed or failed a course; student needs to average 60 or
// more on two tests.
start
  Declarations
    num firstTest
    num secondTest
    num average
    num PASSING = 60
  output "Enter first score or 0 to quit "
  input firstTest
  while firstTest <> 0
    output "Enter second score "
    input secondTest
    average = (firstTest + secondTest) / 2
    output "Average is ", average
    if average >= PASSING then
      output "Pass"
    else
      output "Fail"
    endif
    output "Enter first score or 0 to quit "
    input firstTest
  endwhile
stop
```

You have been diligently testing and debugging your flows and pseudocode programs throughout the course. However, this is the first assignment focused on debugging. Here are some helpful hints to guide you.

The declarations are always a good start to review. should be carefully reviewed to ensure that all variables used are correctly spelled throughout the code.

The conditional statement within the while loop, which checks if 'firstTest' is not equal to 0, dictates that the code block will be continuously executed until the variable 'firstTest' is no longer equal to 0. So lets check.

Just prior to the while loop, there is an output prompting the user to "Enter first score or 0 to quit." However, it appears that there is a missing input statement to store the user's response in the appropriate variable.

A correction should be made by adding an input statement: 'firstTest = input("Enter first score or 0 to quit")'.

We are debugging but we are also testing, so lets check our while statement. We

just collected the firstTest from the user with the input statement, lets assume the user entered in 100. Our while statement is firstTest not equal to 0, our user put in 100, so firstTest is not equal to 0, so we are ok to move to the next statement.

We are currently in the process of debugging and testing our code. Let's review our while statement. We have obtained the value of firstTest from the user using the input statement, with the assumption that the user entered 100. Our while statement checks if firstTest is not equal to 0. Since the user input 100, the condition is met and we can proceed to the next statement.

Let's enhance the readability of the pseudocode by modifying the "not equal" to symbols "<>". The revised line of code will be "while firstTest <> 0".

The subsequent line of code involves formatting the output statement. It is important to keep in mind that pseudocode can have both input and output for clarity.

We will compute the average by using the formula:  $\text{average} = (\text{firstTest} + \text{secondTest}) / 2$ . It is essential to include parentheses around the addition of firstTest and secondTest to ensure accurate calculation before division by 2.

Output looks good, output "Average is ", average

Let's review the IF - ELSE - ENDIF logic. The average has been calculated accurately and we have established that a passing grade is 60. For example, if we input 100 the first time and 80 the second time, our average should be 90. Since the average is greater than or equal to 60, the output will be "Pass". If it is not, the output will be "Fail". Our IF statement has been successfully tested, so we can proceed to the next step.

```
if average >= PASSING then
    output "Pass"
else
    output "Fail"
endif
```

The next step is output "Enter first score or 0 to quit ". since the developer is splitting the input and output in two different lines, we need to have an input statement, add input firstTest.

If the user entered 0 we will exit our while statement because our statement for the while is "while firstTest <> to 0". Our last step is to stop so we are done.

You have successfully achieved the following milestones:

- Formulating and implementing your testing plan
- Methodically executing the test by reviewing each line of code
- Thoroughly verifying the accuracy of calculations and solutions

Well done on effectively strategizing, planning, and debugging the program. Your efforts have been commendable.