|  |  |
| --- | --- |
| Student Name Abdulaziz Elmansouri | Date 9/21/2025 |
| 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:**  Question: What actions could the developer have taken to minimize the number of errors?  **Your Response:** | |
| **Debbugging Exercise: 1**  // This pseudocode is intended to determine whether students have  // passed or failed a course; student needs to average 60 or more on two tests. | |
| Original – with errors | Updated by student |
| Start  Declarations  num firstTest  num secondTest  num average  num PASSING = 60  output "Enter first score or 0 to quit "  while firstTest not equal to 0  output "Enter second score "  input secondTest  average = firstTest + secondTest / 2  ouput "Average is ", average  if average >= PASSING then  output "Pass"  else  output "Fail"  endif  output "Enter first score or 0 to quit "  endwhile  stop | 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 |
| Hints  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: average = (firstTest + 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, ouput "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. | |

|  |  |
| --- | --- |
| **Debug Exercise 02**  // This pseudocode is intended to display employee net pay values.  // All employees have a standard $45 deduction from their checks.  // If an employee does not earn enough to cover the deduction an error message is displayed. | |
| Original – with errors | Updated by student |
| start  Declarations  string name  num hours  num rate  string DEDUCTION = 45  string EOFNAME = "ZZZ"  num gross  num net  output "Enter first name or ", EOFNAME, " to quit"  input name  if name not equal to EOFNAME  output "Enter hours worked for ", name  input hours  output "Enter hourly rate for ", name  input rate  gross = hours \* rate  net = gross - DEDUCTION  while net > 0 then  output "Net pay for ", name, " is ", net  else  output "Deductions not covered. Net is 0."  endwhile  output "Enter next name or ", EOFNAME, " to quit"  input name  endif  output "End of job"  stop | start  Declarations  string name  num hours  num rate  num DEDUCTION = 45  string EOFNAME = "ZZZ"  num gross  num net  output "Enter first name or ", EOFNAME, " to quit"  input name  while name not equal to EOFNAME  output "Enter hours worked for ", name  input hours  output "Enter hourly rate for ", name  input rate  gross = hours \* rate  net = gross - DEDUCTION  if net > 0 then  output "Net pay for ", name, " is ", net  else  output "Deductions not covered. Net is 0."  endif  output "Enter next name or ", EOFNAME, " to quit"  input name  endwhile  output "End of job"  stop |

|  |  |
| --- | --- |
| **Debug Exercise 03**  // This pseudocode is intended to determine whether students have  // passed or failed a course; student needs to average 60 or more on two tests. | |
| Original – with errors | Updated by student |
| start  Declarations  num firstTest  num secondTest  num average  num PASSING = 60  output "Enter first score or 0 to quit "  while firstTest not equal to 0  output "Enter second score "  input secondTest  average = firstTest + secondTest / 2  ouput "Average is ", average  if average >= PASSING then  output "Pass"  else  output "Fail"  endif  output "Enter first score or 0 to quit "  endwhile  stop | start  Declarations  num firstTest  num secondTest  num average  num PASSING = 60  output "Enter first score or 0 to quit "  input firstTest  while firstTest not equal to 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 |
|  |  |

|  |  |
| --- | --- |
| **Debug Exercise 04**  // This pseudocode segment is intended to compute and display  // the cost of home ownership for any number of users  // The program ends when a user enters 0 for the mortgage payment | |
| Original – with errors | Updated by student |
| start  Declarations  num mortgagePayment  num utilities  num taxes  num upkeep  num total  startup()  while mortgagePayment not equal to 0  MainLoop()  endwhile  finishUp()  stop  startUp()  output "Enter your mortgage payment or 0 to quit"  input mtgPayment  return  mainLoop()  output "Enter utilities"  input utilities  output "Enter taxes"  input taxes  output "Enter amount for upkeep"  input upkeep  total = mortgagePayment + utilities + taxes + upkeep  output "Total is ", total  return  finishUp()  output "End of program"  return | start  Declarations  num mortgagePayment  num utilities  num taxes  num upkeep  num total  startUp()  while mortgagePayment not equal to 0  MainLoop()  endwhile  finishUp()  stop  startUp()  output "Enter your mortgage payment or 0 to quit"  input mortgagePayment  return  mainLoop()  output "Enter utilities"  input utilities  output "Enter taxes"  input taxes  output "Enter amount for upkeep"  input upkeep  total = mortgagePayment + utilities + taxes + upkeep  output "Total is ", total  return  finishUp()  output "End of program"  return |