Debug03-01

// 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 // added priming input

while firstTest not equal to 0

output "Enter second score"

input secondTest

average = (firstTest + secondTest) / 2

output "Average is ", average // misspelled output

if average >= PASSING then

output "Pass"

else

output "Fail"

output "Enter first score or 0 to quit "

input firstTest incomplete while loop

endif

endwhile

stop

DEBUG03-02

// 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.

start

Declarations

string name

num hours

num rate

num DEDUCTION = 45 // should be num not string

string EOFNAME = "ZZZ"

num gross

num net

output "Enter first name or ", EOFNAME, " to quit"

input name

while name not equal to EOFNAME // should be a loop instead of selection

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 // should be selection instead of loop

output "Net pay for ", name, " is ", net

else

output "Deductions not covered. Net is 0."

endif // should be selection instead of loop

output "Enter next name or ", EOFNAME, " to quit"

input name

endwhile // should be a loop instead of selection

output "End of job"

stop

DEBUG03-03

// 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.

// This example is modularized.

start

Declarations

string name

string EOFNAME = ZZZZ

housekeeping() // need a name entered before the condition can be evaluated

if name not equal to EOFNAME then //only needs to be a selection

mainLoop()

else

finish()

endif

stop

housekeeping()

output "Enter first name or ", EOFNAME, " to quit "

input name // left out input

return

mainLoop()

Declarations

num hours

num rate

num DEDUCTION = 45

num net

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

return

finish()

output "End of job"

return