Debug04-01

// This pseudocode should create a report that contains an

// apartment complex rental agent's commission. The

// program accepts the ID number and name of the agent who

// rented the apartment, and the number of bedrooms in the

// apartment. The commission is $100 for renting a three-bedroom

// apartment, $75 for renting a two-bedroom apartment, $55 for

// renting a one-bedroom apartment, and $30 for renting a studio

// (zero-bedroom) apartment. Output is the salesperson’s

// name and ID number and the commission earned on the rental.

start

Declarations

num salesPersonID

string salesPersonName

num numBedrooms

num COMM\_3 = 100 // variable type num can't have $ or decimal

num COMM\_2 = 75 // variable type num can't have $ or decimal

num COMM\_1 = 55 // variable type num can't have $ or decimal

num COMM\_STUDIO = 30 // variable type num can't have $ or decimal

num QUIT = 9999

num commissionEarned // undeclared variable added

getReady()

while salesPersonID <> QUIT

detailLoop()

endwhile

finish()

stop

getReady()

output "Enter salesperson ID or ", QUIT, " to quit "

input salespersonID // undeclared variable and input required

return

detailLoop()

output "Enter name "

input salesPersonName

output "Enter number of bedrooms rented "

input numBedrooms

if numBedrooms = 3 then // > 3 will never occur

commissionEarned = COMM\_3

else

if numBedrooms = 2 then // < 2 is 1 bedroom so 2 bedroom rentals never get paid

commissionEarned = COMM\_2

else

if numBedrooms = 1 then // > 1 will include 2 and 3 bedroom rentals

commissionEarned = COMM\_1 // undeclared variable

else

commissionEarned = COMM\_STUDIO // undeclared variable

endif

endif

endif

output salesPersonID, salesPerssonName, commissionEarned // undeclared variable

output "Enter salesperson ID or ", QUIT, " to quit "

input salesPersonID

return

finish()

output "End of report"

return

Debug04-02

// This pseudocode should create a list that describes annual profit

// statistics for a retail store. Input records contain a department

// name (for example, �Cosmetics�) and profits for each quarter for

// the last two years. The program should determine whether

// the profit is higher, lower, or the same

// for this full year compared to the last full year.

start

Declarations

string department

num salesQuarter1ThisYear

num salesQuarter2ThisYear

num salesQuarter3ThisYear

num salesQuarter4ThisYear // repeated 3rd quarter

num salesQuarter1LastYear

num salesQuarter2LastYear

num salesQuarter3LastYear // repeated this year

num salesQuarter4LastYear

num totalThisYear

num totalLastYear

string status

string QUIT = "ZZZZ" // num can not contain characters

housekeeping()

while department <> QUIT

compareProfit()

endwhile

finishUp()

stop

housekeeping()

output "Enter department name or ", QUIT, " to quit "

input department // undeclared variable

return

compareProfit()

getSalesData()

sumSalesData()

if totalThisYear > totalLastYear then // totalThisYear must be greater than totalLastYear

status = "Higher"

else

if totalThisYear < totalLastYear then // can not be equal for Lower status

status = "Lower"

else

status = "Same"

endif

endif

output department, status

output "Enter department name or ", QUIT, " to quit "

input department

return

getSalesData()

output "Enter sales for first quarter this year "

input salesQuarter1ThisYear

output "Enter sales for second quarter this year "

input salesQuarter2ThisYear // wrong variable

output "Enter sales for third quarter this year "

input salesQuarter3ThisYear // wrong variable

output "Enter sales for fourth quarter this year "

input salesQuarter4ThisYear

output "Enter sales for first quarter last year "

input salesQuarter1LastYear

output "Enter sales for second quarter last year "

input salesQuarter2LastYear // wrong variable

output "Enter sales for third quarter last year "

input salesQuarter3LastYear

output "Enter sales for fourth quarter last year "

input salesQuarter4LastYear // wrong variable

return

sumSalesData()

totalThisYear = salesQuarter1ThisYear + salesQuarter2ThisYear +

salesQuarter3ThisYear + salesQuarter4ThisYear // repeated 2nd quarter

totalLastYear = salesQuarter1LastYear + salesQuarter2LastYear +

salesQuarter3LastYear + salesQuarter4LastYear // repeated 2nd quarter

return

finishUp()

output "End of report"

return

Debug04-03

// This pseudocode should determine and output the rental fees for cars.

// Standard cars rent for $65 per day, compacts rent for $40 per day,

// and subcompacts rent for $30 per day. Rentals for at least 7 days

// receive a 20% discount. An error message is displayed if the car type

// is not valid.

start

Declarations

string carType

num days

num STD\_RATE = 65

num COM\_RATE = 40

num SUB\_RATE = 30

num DAYS\_FOR\_DISCOUNT = 7 // should be 7 days

float DISCOUNT\_RATE = 0.20 // num can not have decimals

string QUIT = "ZZZZ"

num rate // declare rate variable

getReady()

while carType <> QUIT

detailLoop()

endwhile

finish()

stop

getReady()

output "Enter car type or ", QUIT, "to quit" // require quotes around output strings

input carType

return

detailLoop()

output "Enter days rented "

input days

if carType = "Standard" then

rate = STD\_RATE

else

if carType = "Compact" then // undeclared variable

rate = COM\_RATE // undeclared variable

else

if carType = "Subcompact" then

rate = SUB\_RATE

else

rate = 0

output "Invalid car type"

endif

endif

endif

if rate <> 0 then // requires then command

if days >= DAYS\_FOR\_DISCOUNT then

rate = rate \* DISCOUNT\_RATE

endif

endif // incomplete loop

output carType, rate // requested output was rate not days

output "Enter car type or ", QUIT, " to quit "

input carType

return

finish()

output "End of program"

return