**11DTP Exercise Tutorial 5 - question 1**

For a particular house, input the length and breadth of a number of rooms. Input is terminated by entering 0 for the length. Calculate and display the area of each room as well as the total area for all the rooms. Calculate and display the cost of carpeting for the house as follows: if the total area exceeds 100 square metres, carpet costs $100 per square metre; else the cost of carpeting is $115 per square metre.

Draw up a test data table for expected data.

Construct a computer program to solve the above task, using the plan given and recording your process of code/test/debug with versions of code

Note that the current building code limits the length or breadth of any room to 20metres.

Improve on your program by creating a test data table for boundary and invalid data, and adding to the code in the next version to construct a robust and flexible program.

**Input**

|  |  |  |
| --- | --- | --- |
| *variableName* | *type* | *How entered* |
| length | float | User - type 0 to quit |
| breadth | User - keyboard |

**Constants /Stored data**

carpetLimit 🡨 100

lowCost 🡨 100

highCost 🡨 115

**Output**

|  |  |  |
| --- | --- | --- |
| *variableName* | *format* | *When it happens* |
| roomArea | 2 dp | For each room |
| houseArea | 2 dp | At end of house data |
| carpetCost | Currency 2 dp |

**Calculations and Processing**

roomArea 🡨 length \* breadth

houseArea 🡨 sum statements

if houseArea > carpetLimit

carpetCost 🡨 houseArea \* lowCost

else

carpetCost 🡨 houseArea \* highCost

**Test data (need to create)**

**Testing plan**

***Version 1***

*#enter data*

declare constants

enter length - type 0 for quit

repeat until length = 0

enter breadth

#processing

enter length - type 0 for quit

#end program

***Version 2***

*#enter data*

*#calculations for rooms*

declare constants

enter length - type 0 for quit

repeat until length = 0

enter breadth

#processing

roomArea 🡨 length \* breadth

display roomArea

enter length - type 0 for quit

#end program

***Version 3***

*#enter data*

*#calculations for rooms*

*#calculations for house area*

declare constants

houseArea 🡨 0

enter length - type 0 for quit

repeat until length = 0

enter breadth

#processing

roomArea 🡨 length \* breadth

display roomArea

houseArea 🡨 houseArea + roomArea

enter length - type 0 for quit

display houseArea

#end program

***Version 4***

*#enter data*

*#calculations for rooms*

*#calculations for house area*

*#calculate carpet cost*

declare constants

houseArea 🡨 0

enter length - type 0 for quit

repeat until length = 0

enter breadth

#processing

roomArea 🡨 length \* breadth

display roomArea

houseArea 🡨 houseArea + roomArea

enter length - type 0 for quit

display houseArea

if houseArea > carpetLimit

carpetCost 🡨 houseArea \* lowCost

else

carpetCost 🡨 houseArea \* highCost

display carpetCost

#end program

**ACHIEVED UP TO HERE**

**Unexpected data Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Data | What is being tested | Expected output | Actual Output |
| Length = 0 | No data entered | Message of no data |  |
| Length = 8, breadth = 5  Length = 10 breadth = 6 | Boundary of houseArea = 100 | houseArea = 100  carpet Cost = 11500 |  |
| Up to here for merit | | | |
| Length = 20 breadth = 20 | Boundary of upper limit for size of rooms | houseArea = 400  carpetCost = 40000 |  |
| Length = -5, 21 breadth = -3, 21 | Invalid data for length and breadth - range of data | Error message and request for more data |  |
| Needed for excellence | | | |

***Version 5***

*#enter data*

*#calculations for rooms*

*#calculations for house area*

*#calculate carpet cost*

*#boundary data*

Flag 🡨 0

declare constants

houseArea 🡨 0

enter length - type 0 for quit

repeat until length = 0

enter breadth

#processing

Flag 🡨 1

roomArea 🡨 length \* breadth

display roomArea

houseArea 🡨 houseArea + roomArea

enter length - type 0 for quit

if flag = 0

display message of no data entered

else

display houseArea

if houseArea > carpetLimit

carpetCost 🡨 houseArea \* lowCost

else

carpetCost 🡨 houseArea \* highCost

display carpetCost

#end program

**MERIT UP TO HERE**

***Version 6***

*#enter data*

*#calculations for rooms*

*#calculations for house area*

*#calculate carpet cost*

*#boundary data*

*#invalid data entered*

Flag 🡨 0

declare constants

houseArea 🡨 0

enter length - type 0 for quit

repeat until length > 0 and length <= 20

error message

enter length - type 0 for quit

repeat until length = 0

enter breadth

repeat until breadth > 0 and breadth <= 20

error message

enter breadth

#processing

Flag 🡨 1

roomArea 🡨 length \* breadth

display roomArea

houseArea 🡨 houseArea + roomArea

enter length - type 0 for quit

if flag = 0

display message of no data entered

else

display houseArea

if houseArea > carpetLimit

carpetCost 🡨 houseArea \* lowCost

else

carpetCost 🡨 houseArea \* highCost

display carpetCost

#end program

#could also make program more robust by allowing data set for more than one house able to be entered (loop within a loop idea)

**EXCELLENCE UP TO HERE**