Flooring Mastery

1. Data layer:3 file types -use enum for dependency purposes
2. Orders Files
3. Named by Orders\_MMDDYYYY.txt
4. CSV
5. Spilt, assign variables to fields, see bank fileIO
6. Includes calculated values
7. Tax files
8. CSV
9. 3 values state abbreviation, state name, sales tax rate
10. Product files
11. CSV
12. Handling Data
13. identify type via interface
14. verify appropriate fields
15. Handle exceptions, display what data input allows
16. Handle inappropriate blanks spaces (split options)
17. Assume more products, states and orders will be created
18. UI
19. Menu options
20. Display order
21. Print from file to console with formatting, see bank option 4-initial note
22. Query by date and load orders.txt file -start spec
23. If !file.exists then display an error
24. Handle with try, catch to avoid exception
25. Display labels for (other fields not labeled in display)
26. Products: [productname]
27. Materials:[material cost]
28. Labor:[laborcost]
29. Tax:[tax amnt ]
30. Total [total cost]
31. Add an order
32. Use streamwriter
33. Be really careful with blank spaces, comma placement
34. Query for each piece of info needed, verify as you go -begin spec
35. Order date -must be in future, parse as datetime, use datetime.compareto to validate
36. Customer name – cannot be blank, **must allow comas**
37. **State – set up states as enum using code**
38. **Product type enum**
39. Area -must be positive decimal min order is 100 ft^2
40. Calculate remaining fields
41. Prompt: place order Y/N write to file if y
42. Create file if first order of the date
43. Edit an order
44. Be sure to resort orders by date as to preserve file order
45. Query for date and order#
46. Handle does not exist
47. Load file for date, for each line in lines Line.contains(ordernumber)
48. Show each piece of data, query user for change, if userinput is null, retain original data
49. Only allow change of customer name, state, product type, or area
50. Recalculate based on changes, display new final order
51. Save, Y/N
52. If yes, replace data in file
53. Remove an order
54. Ask for date and order number
55. If exists, display order, verify removal
56. If order is removed && is only order on file (check count or orders list) delete file
57. Quit
58. Couch all in a while true loop, break if user input =”q” or “Q”’
59. BLL
60. Enum files
61. States
62. ProductTypes
63. File IO – interface with console IO
64. Load orders method
65. Load order method
66. Does file exist (by date) – display error message if not (invalid date)
67. Does file have order# x? display error message if not (invalid order)
68. Display order (called by pretty much everything)
69. Parses through fields of class
70. Remove order method
71. Edit order method
72. Ordering Manager
73. Should load state and product data at start for purposes of calculation
74. Models
75. Order
76. Allow for missing data in files but not user input
77. Customer (allow for default State setting -stretch)
78. State (in list States)
79. Product class
80. Testing
81. Load files
82. Alter files
83. Remove data
84. Add data
85. Calculations
86. Stretch
87. Add state
88. Add product
89. Add customer
90. Change Datetime
91. Record deletions