COMP101 Lab 5: Third Assessed Coursework
Installing Carpet

worth 13% of the final mark
assignment two of seven

Failure to submit this assignment or submission of work that is deemed not to be a reasonable attempt will result in the failure of the module. The completion of the implementation and report as described below will obtain 90% of the marks. To obtain the final 10% students should also complete the Extended Requirements. This may involve doing some additional reading beyond what has been presented in lectures or more complex programming. Only attempt the extended requirements if you are confident with programming.

Learning Outcomes. This addresses the following learning outcomes:

- to be able to implement, compile, test and run Java programmes, comprising more than one class, to address a particular software problem;
- to understand how to include arithmetic operators and constants in a Java program;
- to be able to make use of members of classes found in the Java API (such as the Math class).

Requirements. Imagine that you run a business that installs carpet in people's apartments (or offices).

You are to design, implement and test a Java application program that calculates and prints the floor area of an apartment, and a breakdown of the costs and the total cost to install carpet in an apartment.

We assume that the size of an apartment is (approximately) a rectangle with dimensions (real numbers, measured in metres) given as its length and width. The cost of installing carpet is made up from a cost related to the area of the apartment, plus a labour service charge cost. The cost related to the floor area is £4.75 per square metre. The labour service charge is a fixed amount of £16.50, plus one fourth of the floor area (given in square metres). However, if the area of the apartment is strictly larger than $100m^2$, an extra 10% is added to the labour charge (the total labour charge, including the fixed £16.50 labour charge). On top of the total installation cost (floor area cost plus labour charge), VAT is calculated as 20% of the total cost of the carpet installation.

You should calculate (and print) the floor area (a real number in m^2), the floor area cost, the labour service charge, the total cost before VAT, the VAT, and the total cost. The user of the program will input the length and width of the aparment. You can assume that the values input for the length and width are greater than or equal to zero, i.e. you don't have to check or test for negative inputs from the user.

The calculations are as follows, assuming length and width are in metres.

- $carpetArea = length \times width$
- $carpetAreaCost = floorArea \times 4.75$
- $labourCharge = \begin{cases} 16.50 + (floorArea \div 4) & \text{if } floorArea <= 100 \\ 18.15 + 1.1 * (floorArea \div 4) & \text{if } floorArea > 100 \end{cases}$
- $costBeforeVAT = carpetAreaCost + labourCharge$
- $VAT = costBeforeVAT \times 0.2$
- $totalCost = costBeforeVAT + VAT$

So as stated, you should design, implement and test a Java program which takes as inputs the dimensions of a rectangular apartment, defined in terms of *length* and *width*, and outputs the floor area of the apartment, carpet cost (based on the area), the labour service charge, the cost before VAT, the VAT charged, and total cost (including VAT) of the apartment.

Example. Below is an example of the calculations for length=10.0, and width=8.0.

$$\begin{aligned} carpetArea &= 10.0 \times 8.0 = 80.00m^2 \\ carpetAreaCost &= 80.0 \times 4.75 = 380.00 \\ labourCharge &= 16.50 + (80.0 \div 4) = 36.50 \\ costBeforeVAT &= 380.00 + 36.50 = 416.50 \\ VAT &= 416.50 \times 0.2 = 83.30 \\ totalCost &= 416.50 + 83.30 = 499.80 \end{aligned}$$

The output from this example is given below. Your output should look similar (obviously numbers will differ with different inputs).

```
$ java CarpetCostUser
Please input apartment length (in metres): 10
Please input apartment width (in metres): 8

The apartment has length 10.00 and width 8.00.
Area of apartment: 80.00
Carpet area cost: 380.00
Labour service charges: 36.50
Total before VAT: 416.50
VAT: 83.30
Final Cost: 499.80
```

Another example of the output is below. Note that in this example that the apartment area is bigger than $100m^2$, so the extra labour charges apply here.

```
$ java CarpetCostUser
Please input apartment length (in metres): 12
Please input apartment width (in metres): 10

The apartment has length 12.00 and width 10.00.
Area of apartment: 120.00
Carpet area cost: 570.00
Labour Charges: 51.15
Total before VAT: 621.15
VAT: 124.23
Final Cost: 745.38
```

Extended Requirements. Assume that instead of just one type of carpet, but there are three types as follows:

- basic costing: £4.75 per square metre;
- smart costing: £6.00 per square metre;
- luxury costing: £8.25 per square metre.

You should instead implement a program that has inputs *length* and *width* as before and outputs the area of the apartment, and for *each* type of carpet (basic, smart and luxury), the floor area cost, the labour service charge, the cost before VAT, the VAT charged, and total cost (including VAT) of the apartment.

Hints. Remember to use constants where necessary, and to split your code into self-contained methods. Your calculations should **not** be contained all in one method.

Submission Instructions. Your submission should consist of a report (in pdf format) and implementation files (.java files).

- The report should be a pdf and should consist of

  Requirements: Summary of the above requirements statement.

  Analysis and Design: A short (one paragraph) description of your analysis of the problem including a Class Diagram outlining the class structure for your proposed solution, and pseudocode for methods (you don't need to provide pseudocode for the calculations of floorArea, floorAreaCost, laburCharge, costBeforeVAT, VAT and totalCost). Note that your solution should comprise (at least) two classes: an "application class" and a "target class".

  Important: Make certain that your main application class (the class with the "main" method) is called "CarpetCostUser".

  If you do the extended requirements, make certain that your main application class is called "CarpetCostUserExt".

  Testing: A set of proposed test cases presented in tabular format and expected results, and evidence of the results of testing (the simplest way of doing this is to cut and paste the result of running your test cases into your report or provide a screen shot).

- The implementation should consist of

  Your Java source files, i.e. the relevant .java files, not the class (.class) files.

  Upload your files to

  https://sam.csc.liv.ac.uk/COMP/Submissions.pl

(you will need to log in using your Computer Science username and password).

Mark Scheme. Marks will be awarded for

- Analysis and Design 15%
- Implementation (including style) 65%
- Testing 10%
- Extended Requirements 10%

Please see the module web page for the feedback form.

Note.

- Because submission is handled electronically, ANY FILE submitted past the deadline will be considered (at least) ONE DAY late. Late submissions are subject to the University Policy (see www.csc.liv.ac.uk/department/regulations/practical.html).
- Please make sure your Java classes successfully compile and run on ANY departmental computer system, both Windows and Unix. For this reason the use of powerful IDEs like NetBeans is discouraged.
- Note this is an individual piece of work. Please note the University Guidelines on Academic Integrity (see https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf).