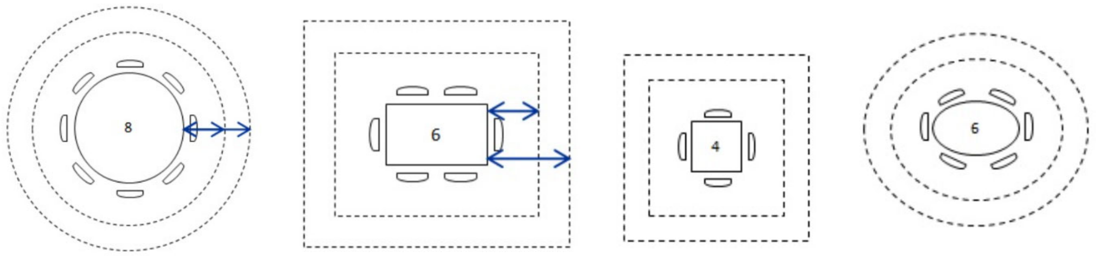
SCHOOL OF COMPUTER SCIENCES
UNIVERSITI SAINS MALAYSIA
ACADEMIC SESSION: 2021/2022

CPT 111 – PRINCIPLES OF PROGRAMMING
Assignment 1 Part A: Hackathon

Dining Table for the Living Room

You are working for an Interior Design company and you are in charge of finding the most appropriate dining table for family dining hall/room.

Based on the dimension provided by the customer of their dining hall/room, find a suitable table that can optimise the space without over-crowding it. The general principle is, to create a comfortable space, there should be about 1m space for the people's movement and chair usage and another 0.6m (minimum) to provide some space so there are some place to roam. You may refer to the diagram below to better depict the spacing description. The biggest table to comfortably fit into the dining hall is considered the best table to choose from.



Round Table Rectangular Table Square Table Oval Table

The possible shape of dining table with the spaces dimension required to make it a comfortable fit based on the living room size.
The first dotted line near the perimeter of the table is the space for people's movement and the chairs while the outer dotted line is the extra space required to roam.

To automate the process of finding suitable table for your customer's space, you want to write a program which is able to at least propose one the best possible tables from a selection of tables given below.

You may design the output so that it will give sufficient information to you on the advantage of the best tables your program has chosen.

Since this algorithm must be able to be used by all prospective houses, which your employer has an agreement with, it must be able to provide **at least one** and if possible the best two solutions based on different living room measurements. However, the program must use metric size (m, cm or mm).

The following are the dining tables available to be chosen by your programme given by the shape of the table and the name to the design:

1. Rectangle Heinrich
   1.3 m x 0.8 m
   6 seaters

2. Rectangle Niklas
   1.5 m x 0.9 m
   6 seaters

3. Rectangle Bertha
   2.1 m x 1.0 m
   8 seaters

4. Rectangle Shade
   3 m x 1 m
   12 seaters

5. Round Nadine
   1.5 m diameter
   8 seaters

6. Round Emma
   1.35 m diameter
   6 seaters

7. Square Lea
   0.9 x 0.9 m
   4 seaters

8. Square Finn
   0.76 x 0.76 m
   2 seaters

9. Oval Stefan
   3 m x 1.3 m
   10 seaters

Assumption: The dimension of the dining hall/room is in length and width and not in any specific order given as input.

You program must have the following features:

i. Interactive – menu to aid user, easy to follow
ii. Meaningful comments in the source codes

Your documentation need to have:

i. The details of the report in the cover:

   Course: CPT111
   Assignment: 1
   Report for: Hackathon 1 Part A
   Group Number: <Your group number **Please refer to the Google Spreadsheet in eLearning**>
   Member List : <Members full name (Matric Number)>
   Lecturer's Name: Dr Nur Hana Samsudin

ii. Table of Content
iii. Description of the question requirements
   a. Analysis of the problem
   b. Identify the specification of the requirements
   c. Design of the program in pseudocode **and** flowchart
   d. Make sure you include inputs, outputs, process and your own constraints and assumptions
iv. The code
v. Sample of cases tested on your program (use print screen with clear print)

Restriction for this Hackathon 1 Part A :

- You **must not** use loop, array, function, pointers or any other topics which only will be covered after Week 4.
- You **must not** use global variable.
- You **must not** use vector, list, queue, or any possible data structure provided by the built-in C++ library.
- You **must not** use <vector>, <stdio>, <list>, <linkedlist>, <queue>, <stack> and any other preprocessor never used before during your lab session. **You may use all the directives** in your programme's pre-processor which you have been exposed to during your class and lab sessions.

How to Submit:

i. You need to compress/zipped all documents into one file. Make sure you have:
   a. The code in .cpp file
   b. Your report in .pdf file
   c. List of the team member in .txt file
ii. Upload your file in the submission link provided in the e-Learning.
iii. Name the folder containing the files in the form of **<Group Number>** only.
iv. There is not specific writing font to use. If you need a relative or comparable size, you may use **Times New Roman** or **Calibri** with **size 11 or 12** for the main content. You may use other font size for sub-title / sub-heading. Please do it in Ms Word or Open Office or Google Doc or any comparable document type and convert to pdf. **Do not write your report in Ms Powerpoint.**

Hackathon Duration

i. This question is released at 16:00 on 19th November 2021 and is due to be submitted at 12:00 midnight on 20th November 2021.
ii. Failure to submit within the timeframe will render you not getting any marks.
iii. No submission outside e-Learning platform will be accepted.

Additional Notes

i. **The team allowing their program or report to be copied** by another team will also get '**F**' **together with the group they shared their program or report** with.
ii. Please refer to rubric to know more about penalties deduction.
iii. **Your lecturer will really not answer your questions if you ghosted your Telegram ID.**