ENGR 102 – Fall 2022
Lab: Topic 8 (team)
50 pts = 100%

Practice
Run 2 examples from Richard L. Halterman "LEARNING TO PROGRAM WITH PYTHON"
- P89, chapter 5.4 Example 1 "Print multiplication table" [20points]
- P91 chapter 5.4 Example "Permutation of ABC" [20 points]

Understand the algorithms

Review all HWs and examples for the Exam

Deliverables:
There are several deliverables for this team assignment. Please submit the following files to Canvas and zyBooks:
Canvas:
- `ascii_clock_planning.pdf 16= 33 Canvas points`
- `ascii_clock_XX.py` where XX represents the initials of the team member

ZyBook
- `ascii_clock.py 8 pts = 17 Canvas points`

Activity #1: Top-Down Design of a Program – team
Following the process described in the lecture, as a team perform a top-down design for the program described below. **AFTER** you plan your program, write the code.

Write a program to take as input from the user a time and display that time using ASCII art. Format your output using the example shown below. Draw each digit 3 characters wide and 5 characters tall, using only spaces and that digit. Draw one space between numbers. Draw the colon 1 character wide. You do NOT need a leading zero or blank space for times earlier than noon.

Example output (using input **12:56**):
```
Enter the time: 12:56

 1  222   555 666
11    2 : 5   6
 1  222   555 666
 1  2   :   5 6 6
111 222   555 666
```

Example output (using input **1:02**):
```
Enter the time: 1:02

 1    000 222
11  : 0 0   2
 1    0 0 222
 1  : 0 0 2
111   000 222
```

BEFORE WRITING ANY CODE, create a document named ascii_clock_planning.pdf that will include parts a-c and g.

a) First, as a team, develop a top-down design for your program. Develop a hierarchy for the individual pieces, breaking each one down into as small of a piece so that the code for that portion of the program is "obvious" (about 5-10 lines of code). Put this hierarchy into your document. You may hand draw your hierarchy or use software to create it.
b) Next, as a team, determine what variables you will use for the main sections of your code. You should decide on what information you will use in more than one "node" of the design. Write a list of these, along with a very brief (one sentence) description of what that variable will store. Note that you do not need to decide on variables that will be used only within one "node" of a program (e.g. you don't need to describe a loop iterator if it is not going to be used outside the loop). Put this into your document.
c) As a team, discuss briefly the test cases you will need to test your code. Write down at least 5 test cases that you plan to use.
d) As a team, create one file with the comments/outline of the code. Share this file among all team members.
e) Next, divide the coding tasks among all team members, so that each person has a different section of the program.
    - Divide the "leaf" nodes among the team members so that each person has approximately the same number of items to implement.
    - If you have done a good job with the top-down design, and in specifying the variables that will carry over from one section to the next, then everyone should be able to write code for just their own section without seeing the other sections of the code!
f) Once each person has written their own separate code, bring the files together as a team, and have one person combine all of the code into one program.
    - You will likely need to thoroughly test and debug your code together as a team at this point. Use the test cases you developed in part c.
g) As a team, write a short summary (a few sentences – at least 75 words for each bullet point)
    - Describing the difficulty with which your team was able to combine the code at the end. Did this provide your team any insight into how the design itself might have been specified more clearly?
    - Describing any benefits and drawbacks you saw into dividing the coding like this. Can you see reasons why this might be a good idea? Can you see reasons why this might be a bad idea?

Things to submit:

1) Your document for parts a, b, c, and g as a single PDF file named ascii_clock_planning.pdf
2) The individual files you produced in part e named ascii_clock_XX.py where XX represents the initials for the team member that wrote the code
3) The single debugged file from part f named ascii_clock.py