Clock Display App in Python

1. Introduction

This task is intended to give you the opportunity to

- practise writing software using the object oriented paradigm
- create a small App with a GUI built using Tkinter
- write a multi-module program that requires more structure and organisation than ones that you have written so far.

2. The task

Create a program with a GUI that displays four working clock faces.

The clock should be an analogue clock with hour, minute and second hands. The clock should be accompanied by a digital time display of hours, minutes and seconds and a separate label for its location e.g. “Oxford”.

The program should display four clock faces in a fixed size window. It must be possible to configure each clock to a distinct world time zone. *It is not necessary for the program to take account of daylight saving time.* The configuration settings which specify the location label for the clock and the time zone of the clock must be held in an ASCII text file which is read at startup. Any other configuration settings may be internal to the program.

The GUI must offer options for About and Exit. The About option will display the version number of the App and author credits. The exit option will close the display window and exit the program.

3. Staging

It is important that an App like this is built and tested in stages so that it works reliably and is easier to test. It also makes it much easier to monitor and review progress.

The work should be undertaken in the following stages:

1. GUI displaying a single clock face with markings (but no hands) and a location label. Any configuration data to be internal to the program.
2. As stage 1 with the addition of the about and exit options to the GUI and a working digital time display for at least 3 different time zones. The clock face should now include stationary hands.
3. As stage 2, but now with working hands (i.e. ones that move as times passes).
4. As stage 3, but with configuration data read from a configuration file.

4. Deliverables

Deliverables are the items that I expect you to produce as a result of your work. This task is delivered in four stages, so for each stage you need to provide:

- A **brief** description of your design that explains how your program is structured and the main features of the processing.
- Suitably **commented** code for that development stage

The delivery timetable will be agreed during lessons as the project progresses.

Appendix 1: Skills

The purpose of this section is to outline some of the key skills you will need in order to complete the task.

1. GUI programming with Tkinter

Look at canvasDemo.py for an example of

- drawing lines, circles and text on a Tkinter canvas – see canvasExtended.py
- updating a canvas at periodic intervals
- setting the title of a Tkinter window

2. Classes – see link below to Using classes in Python

- constructor __init__()
- self
- distinguishing between class and instance variables

3. Time & geometry

- Use of time.localtime() and time.time()
- Determining hand position from time values
- Creating the markings on clock face
- Using instance specific tags to identify graphical elements (e.g. hands)

4. Testing

- Testing small units before integrating them into the App

Useful Links

Tkinter	
https://www.geeksforgeeks.org/python-create-a-digital-clock-using-tkinter/	digital clock with Tkinter
https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/index.html	Tkinter 8.5 reference
https://stackoverflow.com/questions/25753632/tkinter-how-to-use-after-method	The .after() method
https://effbot.org/tkinterbook/canvas.htm	Reference for the Tkinter canvas widget
https://docs.python.org/3/library/tkinter.html	Python standard library doc
Classes	
https://python-textbok.readthedocs.io/en/1.0/Classes.html#defining-and-using-a-class	Using classes in Python
https://python-textbok.readthedocs.io/en/1.0/Introduction_to_GUI_Programming.html	Object-oriented programming in Python
Time processing	
https://www.tutorialspoint.com/python/python_date_time.htm	Getting the current time
https://stackoverflow.com/questions/3168096/getting-computers-utc-offset-in-python	Determining the relationship between UTC and local time
https://www.timeservers.net/cities	World time