EEN118 LAB FIVE

The purpose of this lab is to give you more practice with nice clean structured designs. Make absolutely sure your programs are very clear and obviously correct even to a reader who doesn't know what your design plan was. Grading will be on this basis.

The first two parts should be very fast. Remember the `%` operator. If you find that parts 1 or 2 are taking more than just a few minutes to work out, ask the Lab Guy for a hint. You have probably just not realised what an easy thing to do it really is.

1. *What Time Is It?*

The library contains functions called `get_clock_time` and `get_calendar_date`. They both take no parameters and return an integer as their results. The first tells you the time of day, hours, minutes, and seconds, squeezed into six digits: at 43 seconds past 4:20 p.m. the value would be 162043. The second tells you the date, year, month, and day, squeezed into eight digits: on the 21st of September 2023 the value would be 20230921.

Write a simple program that obtains the current year, month, day, hour, and minute as separate integers. After a little calculation, your program should contain these five lines or something very much like them:

```
print("  year: ");  print(year);   new_line();
print(" month: ");  print(month);  new_line();
print("   day: ");  print(day);    new_line();
print("  hour: ");  print(hour);   new_line();
print("minute: ");  print(minute); new_line();
print("second: ");  print(second); new_line();
```

when run, it should produce output looking something like this:

```
  year: 2023
 month: 9
   day: 21
  hour: 16
minute: 20
second: 43
```

but with the correct numbers of course.

Consider for a moment what very rare problem could possibly occur if you start to run your program just a tiny fraction of a second before midnight. What could your program do guard against that problem?

2. *The Monroe Doctrine.*

Who uses the continental clock? Everyone knows hours are supposed to range from 1 to 12, not zero to 23. Modify your program so that the hour is reduced to the proper 1 to 12 range, but make it also note whether the time is "`a.m.`" or "`p.m.`".

Two little functions, one that takes the raw hour number (0 - 23) and returns the correct 1 - 12 number, and one that also takes the raw hour number but returns the correct "a.m." or "p.m." string, are a very easy way to do this.

```
  year: 2023
 month: 9
   day: 21
period: p.m.
  hour: 4
minute: 20
second: 43
```

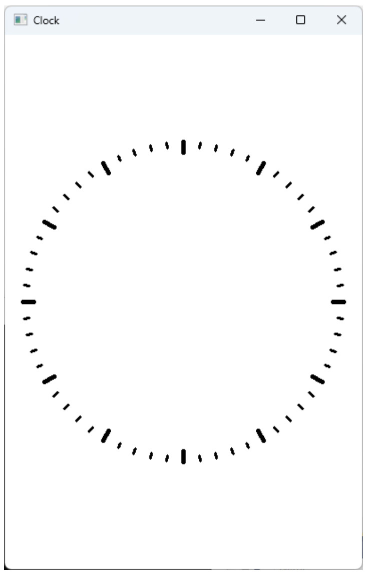
(The Monroe doctrine was the early 19th Century U.S. policy, introduced by President James Monroe, intending to remove the European influence from North America. I'm sure you already knew that.)

3. *A Clock Face.*

Create a window slightly taller than it is wide, and draw tick marks on it for the minutes and hours of a clock face.

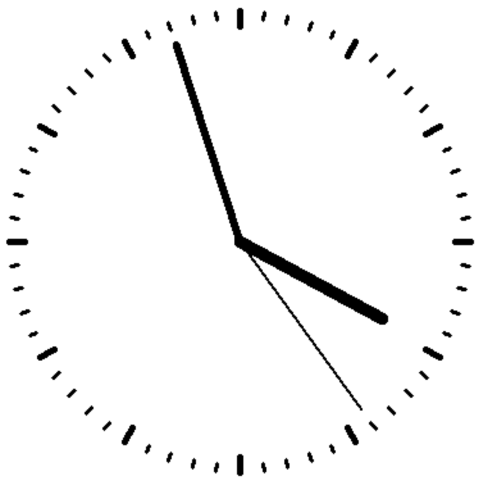
We don't want a solid circle, just the dashes big and small floating there. It's very modern.

The way to draw these marks is a lot like the way to draw an ordinary circle, just a little bit more complicated. Think about how to draw lines radiating all the way out from the centre, then think about how to make them begin further away.



4. *A Whole Clock*

Draw hands on your clock at the correct positions for the current time of day. A little trial-and-error experimentation may be needed to get things just right. We need all three hands, hour, minute, and second.



A hint towards correctness:

If the time is 3:57:24, the hour is 3, but where should the hour hand be pointing?

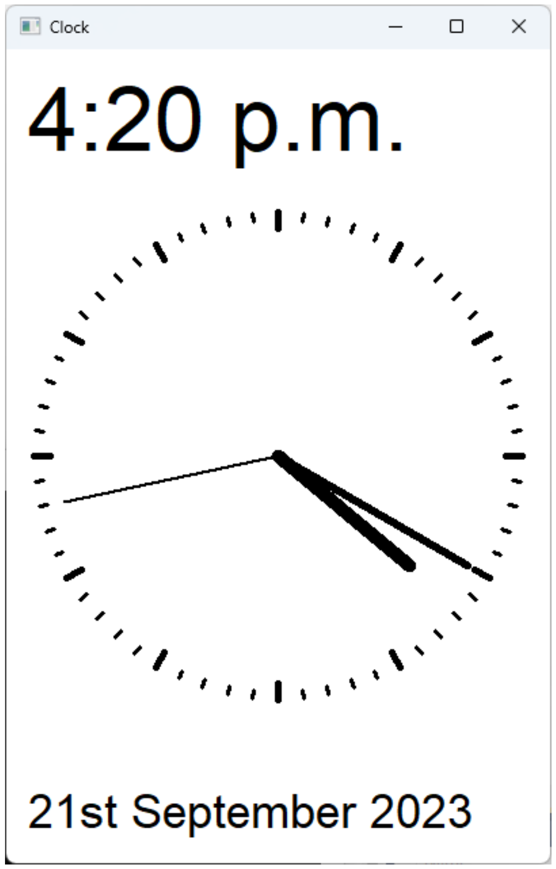
5. *Animate Your Clock.*

Make your clock actually run, so that it keeps updating itself to show the correct time continuously for as long as you let the program run. This works if you repeatedly go through these four steps: look at the time, draw the hands in black, wait for a moderate fraction of a second so it can be seen, and finally draw the hands in white. The function call `wait(0.15)` will cause a pause for 0.15 seconds.

6. *A Complete Product.*

This is why the window isn't square.

Use the space above and below the clock to display the time and date in a nice human-friendly way, like you see below. Don't go overboard with detail, but try to make it look nice.



You have already seen the `write_string` function, but you may not have noticed that it doesn't only display strings, it will happily render `int`s and `double`s: whatever you throw at it. The date was produced by six separate calls to write_string: one for the 21, one for the "st", one for a space, one for the "September", one for another space, and one for the 2023. I used `set_font("Arial", n)`, where n is the desired font size before any `write_string`s. You may see a little flicker occasionally. That can usually be fixed by adjusting the pause length, but don't let it bother you too much.

What are you going to do to ensure that 1405 is displayed as 2:05, and not 2:5?