

HW 1

ENGR 21, Fall 2026.

Solutions

1 Writing simple programs for the Circuit Playground Express

Using only the features of Python that have been discussed so far,

1. Write a program for the Circuit Playground Express that plays a tone of 440 Hz for one-second intervals repeatedly until a cell phone light is shined on it, at which point the program should terminate.

 Solution

```
while cpx.light < 100:  
    cpx.play_tone(440,1)
```

2. Write a program for the Circuit Playground Express that plays a tone of 440 Hz for one-second intervals repeatedly. When it is brought close to a warm object (warmer than room temperature), it stops playing this tone and instead plays a high-frequency tone of 880 Hz for a half second. After this, the program should terminate.

 Solution

```
while cpx.temperature < 31:  
    cpx.play_tone(440,1)  
cpx.play_tone(880,0.5)
```

2 Truth Tables

We have seen that the truth table for the **and** and **or** operators was as given in [lecture](#).

i Note

It is possible to use logic to determine the truth tables for more complicated operators than **and** and **or**, but here you should simply check all possibilities using Python.

2.1 For a binary operator

Use this info to construct a truth table for the following binary operator.

(a and b) or a

💡 Solution

Variable a	Variable b	Result of (a and b) or a
True	True	True
True	False	True
False	True	False
False	False	False

2.2 For an operator with three inputs

Construct a truth table for the following operator.

(a or b) and c

💡 Solution

There are $2^3 = 8$ rows.

a	b	c	(a or b) and c
True	True	True	True
True	True	False	False
True	False	True	True
True	False	False	False
False	True	True	True
False	True	False	False
False	False	True	False
False	False	False	False

3 While loops

Determine the behavior of the Circuit Playground Express when the following code is run on it. Give your answer in words.

3.1 No inputs

```
1. while True:
    cpx.play_tone(440,1)
    cpx.play_tone(880,0.5)
```

 Solution

This will play 440 Hz and will never play 880 Hz.

```
2. while True:
    cpx.play_tone(880,0.5)
    cpx.play_tone(440,1)
```

 Solution

This will play 880 Hz and will never play 440 Hz.

```
3. while True:
    cpx.play_tone(880,0.5)
    cpx.play_tone(440,1)
```

 Solution

This will play 880 Hz for half second, followed by 440 Hz for one second, followed by 880 Hz for half second, and will continue alternating between the two for ever.

```
4. while True:
    cpx.play_tone(440,1)
    cpx.play_tone(880,0.5)
```

 Solution

This will play 440 Hz for one second, followed by 880 Hz for half second, followed by 440 Hz for one second, and will continue alternating between the two for ever.

```
5. cpx.play_tone(440,1)
   while True:
       cpx.play_tone(880,0.5)
```

 Solution

This will play 440 Hz for one second, followed by infinitely-repeating 880 Hz notes for half second intervals. 440 Hz will only play once.

```
6. cpx.play_tone(440,1)
   while False:
       cpx.play_tone(880,0.5)
```

 Solution

This will play 440 Hz for one second, and then nothing more will happen.

```
7. cpx.play_tone(440,1)
   while False:
       cpx.play_tone(880,0.5)
       cpx.play_tone(440,1)
```

 Solution

This will play 440 Hz for one second, followed by another 1-second note at 440 Hz. 880 Hz will never play.

3.2 With inputs

For the following questions, determine what the board will do under the different relevant possibilities. For example,

- if a flashlight is shining on it vs. not
- if the temperature sensor is above/below a threshold
- if a certain pin is touched or not touched.

```
1. while not (cpx.touch_A1 and cpx.touch_A5):
    cpx.play_tone(440,1)
    cpx.play_tone(880,0.5)
```

 Solution

This will play 440 Hz for one second intervals repeatedly. If only pin A1 is touched, nothing will happen, and if only pin A5 is touched, nothing will happen. But if both of these pins are touched simultaneously, the 440 Hz notes will terminate, the board will make one half-second note at 880 Hz, and then the program will terminate.

```
2. while not (cpx.touch_A1 and cpx.temperature > 30):
    cpx.play_tone(440,1)
```

```
cpx.play_tone(880,0.5)
```

💡 Solution

This will play 440 Hz for one second intervals repeatedly. If only pin A1 is touched, nothing will happen as long as the temperature sensor reads less than 30 degrees Celsius. If the temperature sensor is brought near something warm while pin A1 is being touched, the 440 Hz note will cease and the board will make one half-second note at 880 Hz, and then the program will terminate.

```
3. while not (cpx.touch_A1 or cpx.temperature > 30):  
    cpx.play_tone(440,1)  
    cpx.play_tone(880,0.5)
```

💡 Solution

This will play 440 Hz for one second intervals repeatedly. It will stop doing so if one of two conditions is met: **Either** pin A1 is touched, **or** the temperature is raised above 30 degrees Celsius, then the 440 Hz note will cease and the board will make one half-second note at 880 Hz, and then the program will terminate.

```
4. cpx.play_tone(440,1)  
while cpx.light < 100:  
    cpx.play_tone(880,0.5)  
    cpx.play_tone(440,1)
```

💡 Solution

This will play 440 Hz for one second. After that, the program will continue to beep at 880 Hz for half-second intervals repeatedly if it is in low light conditions. If a flashlight is shined on it, the board will produce one second-long note at 440 Hz, after which the program will terminate.

4 Nested while loops

1. Explain the difference between the following programs.

```
while True:  
    cpx.play_tone(880,0.5)  
    while True:  
        cpx.play_tone(440,1)
```

```
while True:  
    cpx.play_tone(880,0.5)
```

```
while True:
    cpx.play_tone(440,1)
```

💡 Solution

With the first program, the Circuit Playground Express will play a note of 880 Hz once, and then will subsequently keep repeating a note of 440 Hz for all eternity. With the second program, the Circuit Playground Express will repeat 880 Hz notes for all eternity. It will never play a 440 Hz note.

2. For each of the following parts, assume that the Circuit Playground Express has been running the following code for a few seconds before you carry out your intervention. (i.e., a, b, c and d are **not** applied sequentially.)

```
while not cpx.touch_A7:
    cpx.play_tone(880,0.5)
    while not cpx.touch_A5:
        cpx.play_tone(440,1)
```

- What happens if pin A5 is tapped ?
- What happens if pin A5 is touched for ten seconds ? Then, when you release it, what happens?
- What happens if pin A7 is touched while the speaker is beeping at 440 Hz ?
- What happens if both pin A5 and A7 are pressed for a few seconds?

💡 Solution

If left for a long time, the Circuit Playground Express will be playing notes of 440 Hz repeatedly.

- If pin A5 is tapped, the program will exit the inner loop and go back to the outer loop, playing a note of 880 Hz once. Then, it will go back to playing notes of 440 Hz repeatedly.
- If pin A5 is touched for ten seconds, the program will exit the inner loop and play notes of 880 Hz repeatedly, since the condition `not cpx.touch_A5` will be `False`, so the inner loop won't run, but the condition `not cpx.touch_A7` will be `True`, so the outer loop will keep running.
- Nothing happens if pin A7 is touched while the speaker is beeping at 440 Hz.
- The program will terminate if both pins are pressed for a few seconds.

3. If the Circuit Playground Express has been running the following code for a few seconds, what happens if pin A5 is touched?

```
while not cpx.touch_A7:
    cpx.play_tone(880,0.5)
    while not cpx.touch_A5:
        cpx.play_tone(440,1)
    while True:
        cpx.play_tone(220,1)
```

 Solution

After a few seconds when no pin is touched, the Circuit Playground Express will be playing a note of 440 Hz repeatedly. If pin A5 is touched, the program will enter into an infinite loop of playing 220 Hz tones for ever.