

SWINBURNE UNIVERSITY OF TECHNOLOGY

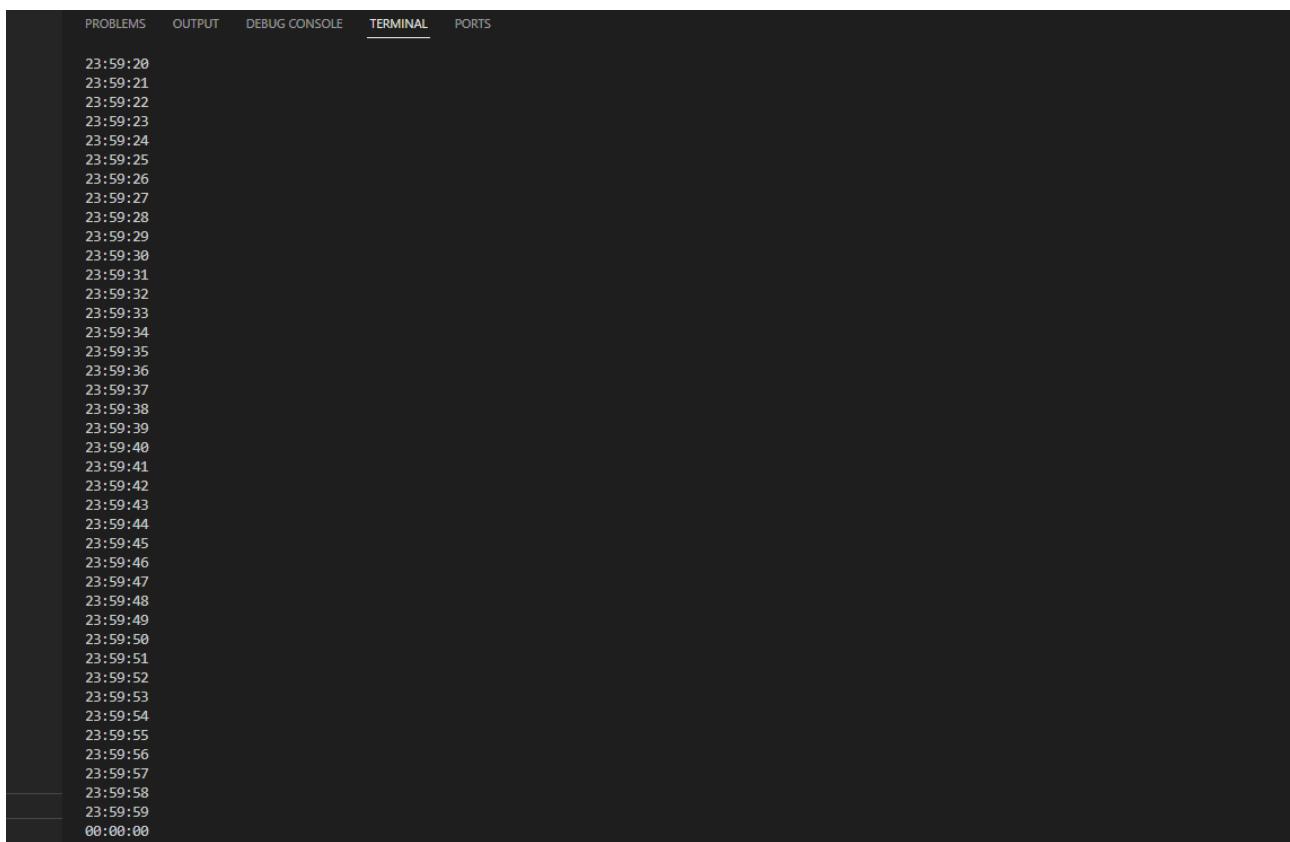
COS20007 OBJECT ORIENTED PROGRAMMING

Clock in Another Language

PDF generated at 17:31 on Thursday 21st September, 2023

```
1 Counter.py
2 class Counter:
3     def __init__(self, name):
4         self._name = name
5         self._count = 0
6         # initialise counter with a name
7
8     @property
9     def count(self):
10        return self._count
11
12    def increment(self):
13        self._count += 1
14        # count value increments by 1
15
16    def reset(self):
17        self._count = 0
18        # count value reset to 0
19
20
21
22 clock.py
23
24 from counter import Counter
25
26 class Clock:
27     def __init__(self):
28         self._hours = Counter("hours")
29         self._minutes = Counter("minutes")
30         self._seconds = Counter("seconds")
31
32     def tick(self):
33         self._seconds.increment()
34         # increment of seconds
35
36         if self._seconds.count == 60:
37             self._seconds.reset()
38             self._minutes.increment()
39             # as seconds reach 60, minutes raise by 1
40
41         if self._minutes.count == 60:
42             self._minutes.reset()
43             self._hours.increment()
44             # as minutes reach 60, hours raise by 1
45
46         if self._hours.count == 24:
47             self._hours.reset()
48             # clock resets once hours reach 24
49
50     def read_time(self):
51         # format time string
52         display_time = "{:02d}:{:02d}:{:02d}".format(
53             self._hours.count, self._minutes.count, self._seconds.count
```

```
54         )
55     return display_time
56
57     def reset(self):
58         self._seconds.reset()
59         self._minutes.reset()
60         self._hours.reset()
61
62
63
64 main.py
65
66
67 from clock import Clock
68
69 def main():
70     clock = Clock()
71
72     print("Digital Clock")
73
74     # 86400 seconds = 24 hours
75     for _ in range(86400):
76         clock.tick()
77         print(clock.read_time())
78
79     clock.reset()
80
81
82 if __name__ == "__main__":
83     main()
84     # entry point of the program
85
```



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are tabs labeled PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The main area of the terminal contains a list of timestamps, each on a new line, starting from 23:59:20 and ending at 00:00:00. The timestamps are spaced evenly, indicating one-second intervals.

```
23:59:20
23:59:21
23:59:22
23:59:23
23:59:24
23:59:25
23:59:26
23:59:27
23:59:28
23:59:29
23:59:30
23:59:31
23:59:32
23:59:33
23:59:34
23:59:35
23:59:36
23:59:37
23:59:38
23:59:39
23:59:40
23:59:41
23:59:42
23:59:43
23:59:44
23:59:45
23:59:46
23:59:47
23:59:48
23:59:49
23:59:50
23:59:51
23:59:52
23:59:53
23:59:54
23:59:55
23:59:56
23:59:57
23:59:58
23:59:59
00:00:00
```