Swinburne University of Technology

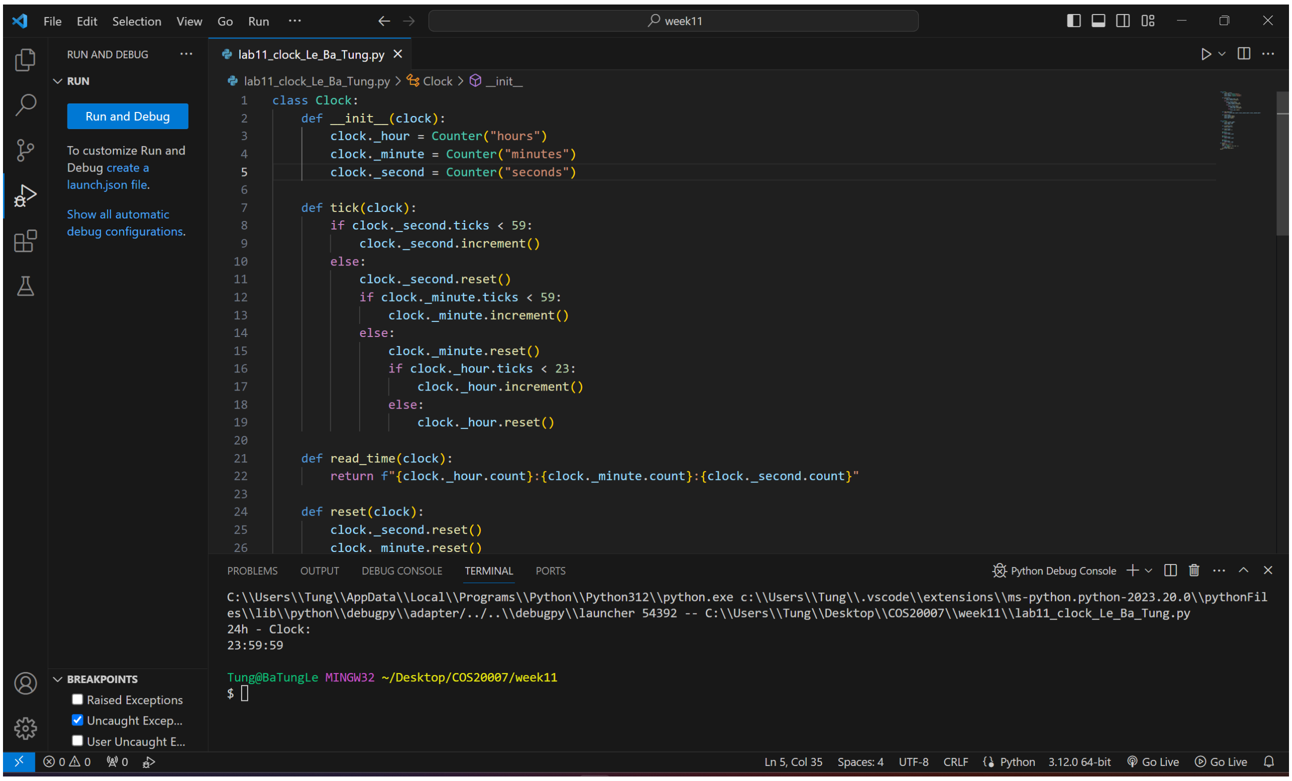
COS20007 Object Oriented Programming

Clock in Another Language

PDF generated at 20:48 on Tuesday 21st November, 2023

Produced by OOP Submission Formatter

```
class Clock:
    def __init__(clock):
        clock._hour = Counter("hours")
        clock._minute = Counter("minutes")
        clock._second = Counter("seconds")

    def tick(clock):
        if clock._second.ticks < 59:
            clock._second.increment()
        else:
            clock._second.reset()
            if clock._minute.ticks < 59:
                clock._minute.increment()
            else:
                clock._minute.reset()
                if clock._hour.ticks < 23:
                    clock._hour.increment()
                else:
                    clock._hour.reset()

    def read_time(clock):
        return f"{clock._hour.count}:{clock._minute.count}:{clock._second.count}"

    def reset(clock):
        clock._second.reset()
        clock._minute.reset()
        clock._hour.reset()


class Counter:
    def __init__(clock, name):
        clock._name = name
        clock._count = 0

    def increment(clock):
        clock._count += 1

    def reset(clock):
        clock._count = 0

    @property
    def ticks(clock):
        return clock._count

    @property
    def count(clock):
        return clock._count

    @property
    def name(clock):
        return clock._name
def main():
    print("24h - Clock:")
    time = Clock()
    for i in range(24 * 60 * 60 - 1):
        time.tick()
    print(time.read_time())
main()
```



File Edit Selection View Go Run ···
week11
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Excep...
□ User Uncaught E...
lab11_clock_Le_Ba_Tung.py
lab11_clock_Le_Ba_Tung.py > Clock > __init__
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 5, Col 35 Spaces: 4 UTF-8 CRLF Python 3.12.0 64-bit Go Live Go Live

```python
class Clock:
    def __init__(clock):
        clock._hour = Counter("hours")
        clock._minute = Counter("minutes")
        clock._second = Counter("seconds")

    def tick(clock):
        if clock._second.ticks < 59:
            clock._second.increment()
        else:
            clock._second.reset()
            if clock._minute.ticks < 59:
                clock._minute.increment()
            else:
                clock._minute.reset()
                if clock._hour.ticks < 23:
                    clock._hour.increment()
                else:
                    clock._hour.reset()

    def read_time(clock):
        return f"{clock._hour.count}:{clock._minute.count}:{clock._second.count}"

    def reset(clock):
        clock._second.reset()
        clock._minute.reset()
```

```
C:\\Users\\Tung\\AppData\\Local\\Programs\\Python\\Python312\\python.exe c:\\Users\\Tung\\.vscode\\extensions\\ms-python.python-2023.20.0\\pythonFil
es\\lib\\python\\debugpy\\adapter/../..\\debugpy\\launcher 54392 -- C:\\Users\\Tung\\Desktop\\COS20007\\week11\\lab11_clock_Le_Ba_Tung.py
24h - Clock:
23:59:59

Tung@BaTungLe MINGW32 ~/Desktop/COS20007/week11
$ 
```