Clock.py

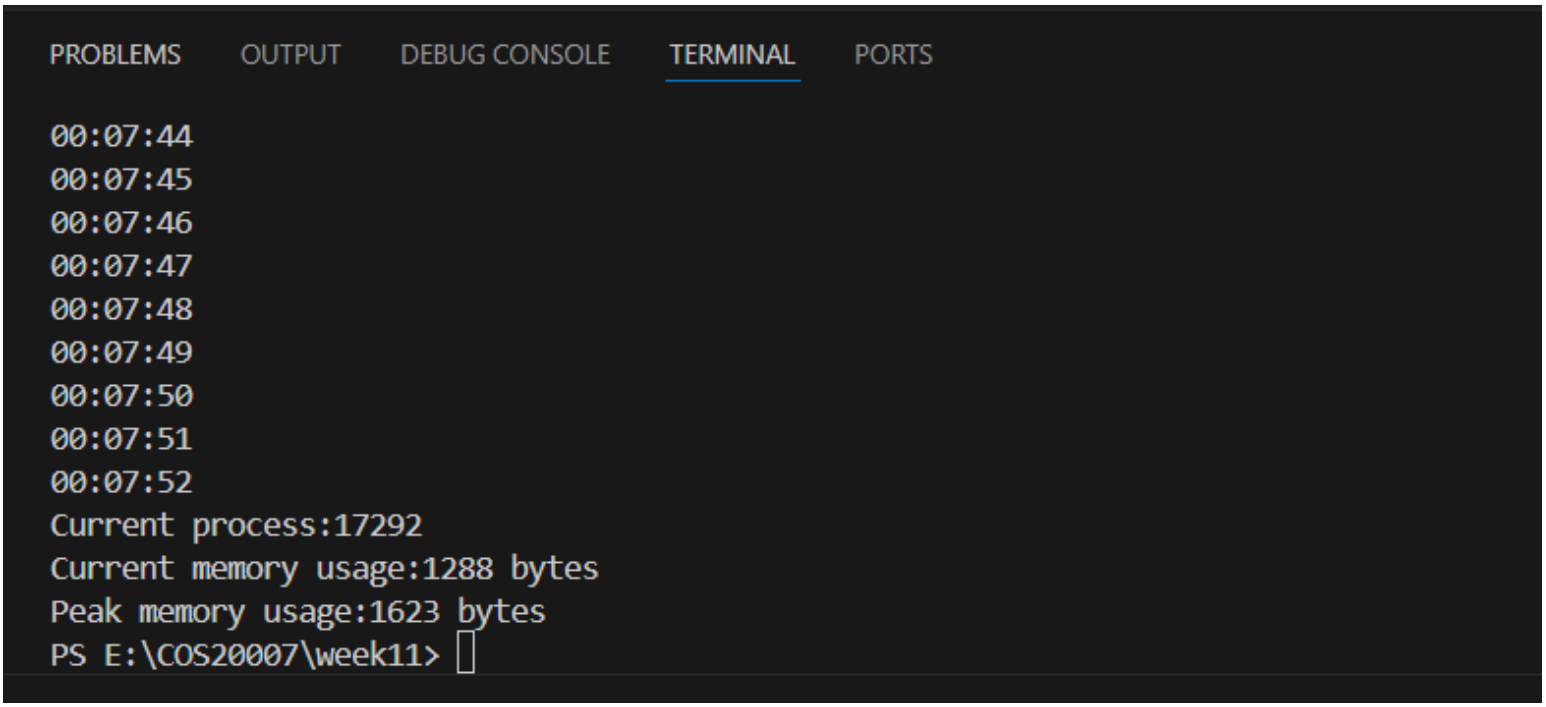
```python
from Counter import Counter
class Clock:
    def __init__(self):
        self._hour=Counter("Hour")
        self._min=Counter("Minute")
        self._sec=Counter("Second")

    def tick(self):
        if self._sec.ticks<59:
            self._sec.increment()
        else:
            self._sec.reset()
            if self._min.ticks<59:
                self._min.increment()
            else:
                self._min.reset()
                if self._hour.ticks<11:
                    self._hour.increment()
                else:
                    self._hour.reset()

    def reset(self):
        self._hour.reset()
        self._min.reset()
        self._sec.reset()

    @property
    def clock_time(self):
        return f"{self._hour.ticks:02}:{self._min.ticks:02}:{self._sec.ticks:02}"

```

Counter.py

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

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

    def reset(self):
        self._count=0

    @property
    def name(self):
        return self._name

    @name.setter
    def name(self, value):
        self._name=value

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

    @ticks.setter
    def ticks(self, value):
        self._count=value

```

Program.py

```python
import os
import tracemalloc
from Counter import Counter
from Clock import Clock

def main():
    tracemalloc.start()
    myclock= Clock()
    for i in range(473):  # 43200
        print(myclock.clock_time)
        myclock.tick()

    current_memory, peak_memory = tracemalloc.get_traced_memory()
    tracemalloc.stop()

    print(f"Current process:{os.getpid()}")
    print(f"Current memory usage:{current_memory} bytes")
    print(f"Peak memory usage:{peak_memory} bytes")

if __name__ == "__main__":
    main()

```



C# memory usage and execution time

```
00:07:52
00:07:53
Current process: System.Diagnostics.Process (Task3_1P)
Physical memory usage: 29278208 bytes
Peak physical memory usage 29278208 bytes

E:\COS20007\week3\Task3_1P\Task3_1P\bin\Debug\net8.0\Task3_1P
.exe (process 32072) exited with code 0 (0x0).
To automatically close the console when debugging stops, enab
le Tools->Options->Debugging->Automatically close the console
 when debugging stops.
Press any key to close this window . . .
```

Python memory usage and execution time

```
00:07:44
00:07:45
00:07:46
00:07:47
00:07:48
00:07:49
00:07:50
00:07:51
00:07:52
Current process:17292
Current memory usage:1288 bytes
Peak memory usage:1623 bytes
PS E:\COS20007\week11>
```