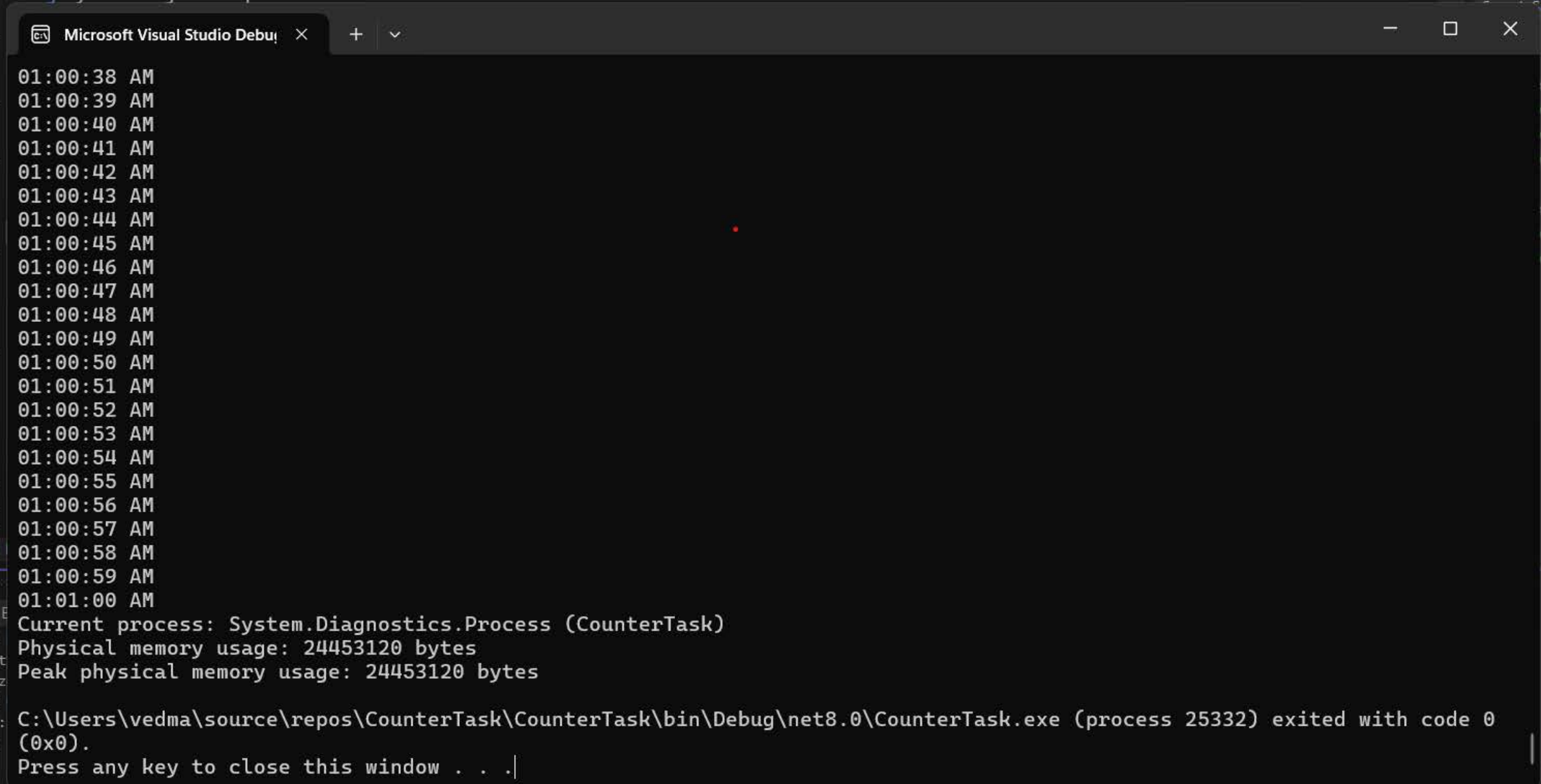
~\OneDrive - Swinburne University\sem2\OOP\Wewk11\11.1p.py

```python
import psutil
import os

class CounterClass:
    def __init__(self, names):
        self._counts = 0
        self._names = names

    @property
    def getname(self):
        return self._names

    @property
    def setname(self, values):
        self._names = values

    @property
    def counterticks(self):
        return self._counts

    def incrementtick(self):
        self._counts += 1

    def resettick(self):
        self._counts = 0


class ClockClass:
    def __init__(self):
        self._sec = CounterClass("sec")
        self._min = CounterClass("min")
        self._hrs = CounterClass("hrs")

    def increment_the_clock(self):
        self._sec.incrementtick()
        if self._sec.counterticks == 60:
            self._sec.resettick()
            self._min.incrementtick()

        if self._min.counterticks == 60:
            self._min.resettick()
            self._hrs.incrementtick()

        if self._hrs.counterticks == 24:
            self._hrs.resettick()

    def show(self):
```

```
48          return f"{self._hrs.counterticks:02d} : {self._min.counterticks:02d} :
    {self._sec.counterticks:02d}"
49
50      def memory_usage(self):
51          process = psutil.Process(os.getpid())
52          mem_info = process.memory_info()
53          return mem_info.rss  # Return the RSS (Resident Set Size) memory usage
54
55
56
57  if __name__ == "__main__":
58      newClock = ClockClass()
59
60      for i in range(86400):
61          newClock.increment_the_clock()
62          print(newClock.show())
63
64      # Print memory usage after the clock has been incremented for 86400 ticks
65      print(f"Memory Usage: {newClock.memory_usage() / (1024 * 1024):.2f} MB")  # Print memory
    usage in MB
66
```

```
1  using System.Diagnostics;
2
3  namespace CounterTask
4  {
5      internal class Program
6      {
7          static void Main(string[] args)
8          {
9              Clock myClock = new Clock(4); // Example with 12-hour format
10
11             for (int i = 0; i < 3660; i++) // Tick the clock 3661 times
12             {
13                 myClock.Tick();
14                 Console.WriteLine(myClock.GetTime());
15             }
16
17             Process proc = Process.GetCurrentProcess();
18             Console.WriteLine("Current process: {0}", proc.ToString());
19             Console.WriteLine("Physical memory usage: {0} bytes",
                 proc.WorkingSet64);
20             Console.WriteLine("Peak physical memory usage: {0} bytes",
                 proc.PeakWorkingSet64);
21         }
22     }
23 }
24
```


Microsoft Visual Studio Debu
01:00:38 AM
01:00:39 AM
01:00:40 AM
01:00:41 AM
01:00:42 AM
01:00:43 AM
01:00:44 AM
01:00:45 AM
01:00:46 AM
01:00:47 AM
01:00:48 AM
01:00:49 AM
01:00:50 AM
01:00:51 AM
01:00:52 AM
01:00:53 AM
01:00:54 AM
01:00:55 AM
01:00:56 AM
01:00:57 AM
01:00:58 AM
01:00:59 AM
01:01:00 AM
Current process: System.Diagnostics.Process (CounterTask)
Physical memory usage: 24453120 bytes
Peak physical memory usage: 24453120 bytes
C:\Users\vedma\source\repos\CounterTask\CounterTask\bin\Debug\net8.0\CounterTask.exe (process 25332) exited with code 0 (0x0).
Press any key to close this window . . .

EXPLORER

OPEN EDITORS
11.1p.py C:\Users\vedma\OneDriv...
Extension: Print

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

Clone Repository

You can open a remote repository or pull request without cloning.

Open Remote Repository

To learn more about how to use Git and source control in VS Code read our docs.

OUTLINE
TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
23 : 59 : 17
23 : 59 : 18
23 : 59 : 19
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
Memory Usage: 15.37 MB
PS C:\Users\vedma>
```

Code
Code