

# Counter.py

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

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

    def reset(self):
        self._count = 0
        return self._count

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

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

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

# Program.py

```
from Clock import Clock

if __name__ == "__main__":
    clock = Clock()
    clock.start_clock(56)
```

# Clock.py

```
import time
from Counter import Counter

class Clock:
    def __init__(self):
        self._hours = Counter("Hours")
        self._minutes = Counter("Minutes")
        self._seconds = Counter("Seconds")

    @property
    def hours(self):
        return self._hours.tick

    @property
    def minutes(self):
        return self._minutes.tick

    @property
    def seconds(self):
        return self._seconds.tick

    def tick(self):
        if self._seconds.tick < 59:
            self._seconds.increment()
        elif self._minutes.tick < 59:
            self._seconds.reset()
            self._minutes.increment()
        elif self._hours.tick < 23:
            self._seconds.reset()
            self._minutes.reset()
            self._hours.increment()
        else:
            self._hours.reset()
            self._minutes.reset()
            self._seconds.reset()

    def reset(self):
        self._hours.reset()
        self._minutes.reset()
        self._seconds.reset()

    def print_time(self):
        current_time =
f"{self._hours.tick:02}:{self._minutes.tick:02}:{self._seconds.tick:02}"
        print(current_time)
        return current_time

    def start_clock(self, seconds):
        for _ in range(seconds):
            time.sleep(1)
            self.tick()
            self.print_time()
```

# Output Screenshot

