```
function Clock({ template }) {

  let timer;

  function render() {
    let date = new Date();

    let hours = date.getHours();
    if (hours < 10) hours = '0' + hours;

    let mins = date.getMinutes();
    if (mins < 10) mins = '0' + mins;

    let secs = date.getSeconds();
    if (secs < 10) secs = '0' + secs;

    let output = template
      .replace('h', hours)
      .replace('m', mins)
      .replace('s', secs);

    console.log(output);
  }

  this.stop = function () {
    clearInterval(timer);
  };

  this.start = function () {
    render();
    timer = setInterval(render, 1000);
  };

}

let clock = new Clock({ template: 'h:m:s' });
clock.start();
```

Answer below questions based on Clock constructor function.

- Also add a line of code that will stop the clock after 10 ticks

```
setTimeout(()=>clock.stop(), 10000);
```

- What is the inner function of the constructor function?
  render
- What is the local variable of the constructor function?
  timer
- What is the clock “interface” returned by the constructor function?
  An object of Clock
- What are the closures?
  No
- What are the private variables and functions?
  timer, & render
- What are the public methods?
  stop, & start

```
class Clock {
  constructor({ template }) {
    this.template = template;
  }

  render() {
    let date = new Date();

    let hours = date.getHours();
    if (hours < 10) hours = '0' + hours;

    let mins = date.getMinutes();
    if (mins < 10) mins = '0' + mins;

    let secs = date.getSeconds();
    if (secs < 10) secs = '0' + secs;

    let output = this.template
      .replace('h', hours)
      .replace('m', mins)
      .replace('s', secs);

    console.log(output);
  }

  stop() {
    clearInterval(this.timer);
  }

  start() {
    this.render();
    this.timer = setInterval(() => this.render(), 1000);
  }
}
let clock = new Clock({ template: 'h:m:s' });
clock.start();
```

Answer below questions based on Clock constructor function.

- Also add a line of code that will stop the clock after 10 ticks

```
setTimeout(()=>clock.stop(), 10000);
```

- What is the inner function of the constructor function?
  no
- What is the local variable of the constructor function?
  no
- What is the clock “interface” returned by the constructor function?
  An object of Clock
- What are the closures?
  no
- What are the private variables and functions?
  no
- What are the public methods?
  stop, start, & render
- How does this example illustrate that a JavaScript class is really a function and not an object?
  I don’t see the actual relationship here. Discuss in the class.