

Portfolio element – JavaScript

Unit	Programming languages: principles and design (6G6Z1110) Programming languages – SE frameworks (6G6Z1115)
Lecturer	Rob Frampton
Week	7
Portfolio element	JavaScript (15% of coursework)

Assignment

In this assignment you will implement a JavaScript class which queues up arithmetic operations, then executes them starting from a particular number. For example, you may queue up the following operations:

- Add 2
- Multiply by 4
- Add 10

Then, if you execute these operations starting from the number 2, you would receive the result 26, while if you execute from the number -2, you would receive the result 10.

You should implement a class called `ArithmeticTaskRunner` with the following:

- An instance variable named `tasks` which is initialised to an empty array upon creation.
- A method named `addNegationTask` which adds an anonymous function to the `tasks` array. This anonymous function should take one argument, `x`, and return the negation, `-x`.
- A method named `addAdditionTask` which takes a single argument `y`, and adds an anonymous function to the `tasks` array. This anonymous function should take one argument, `x`, and return the result `x + y`.
- A method named `addMultiplicationTask` which takes a single argument `y`, and adds an anonymous function to the `tasks` array. This anonymous function should take one argument, `x`, and return the result `x * y`.
- A read-only accessor named `taskCount` which returns the number of queued tasks.
- A method named `execute`, which takes a single argument named `startValue`. If omitted, `startValue` defaults to zero. Starting at `startValue`, this method should iterate over the `tasks` array executing each function on the current value. It then returns the resulting number after all arithmetic operations have been executed.

Example

You can test your code by observing the output when running the following commands:

```
> .load ArithmeticTaskRunner.js
...
> let taskRunner = new ArithmeticTaskRunner()
undefined
> taskRunner.addAdditionTask(10)
undefined
> taskRunner.addNegationTask()
undefined
> taskRunner.addMultiplicationTask(0.5)
undefined
> taskRunner.execute()
```

Portfolio element – JavaScript

```
-5  
> taskRunner.execute(10)  
-10  
> taskRunner.taskCount  
3
```

Here is another example:

```
> let taskRunner = new ArithmeticTaskRunner()  
undefined  
> taskRunner.addAdditionTask(2)  
undefined  
> taskRunner.addMultiplicationTask(4)  
undefined  
> taskRunner.addAdditionTask(10)  
undefined  
> taskRunner.execute(2)  
26  
> taskRunner.execute(-2)  
10
```

You should also test that `taskCount` is read-only:

```
> taskRunner.taskCount  
3  
> taskRunner.taskCount = 10  
10  
> taskRunner.taskCount  
3
```

Note: if you have enabled strict mode, trying to set `taskCount` will cause an error to be thrown. This is also fine.

Submission

You must submit a file named “ArithmeticTaskRunner.js” through Moodle. Submission link is available under Week 7 section.