

Smalltalk Lecture 4

Assignment Project Exam Help

Add WeChat powcoder

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Assignment Project Exam Help

Fraction class

Add WeChat powcoder

Number subclass: #MyFraction.

MyFraction instanceVariableNames: 'num den'.

MyFraction extend

num [^ num]

den [^ den]

num: n den: d [num = n. den = d]

]

f1 := MyFraction new num: 10 den: 4.

f2 := MyFraction new num: 15 den: 6.

f3 := MyFraction new num: -10 den: 4.

f4 := MyFraction new num: 15 den: -6.

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

printOn:

Add WeChat powcoder

MyFraction extend [

printOn: aStream [

num printOn: aStream.

aStream nextPut: \$/.

den printOn: aStream.

aStream cr]

]

f1 printOn: Transcript.

f2 printOn: Transcript.

f3 printOn: Transcript.

f4 printOn: Transcript.

10/4

15/6

-10/4

15/-6

Assignment Project Exam Help

+ and -

Add WeChat powcoder

MyFraction extend [

+ f [^ MyFraction new num: (num * f den) + (den * f num)
den: (den * f den)]

- f [^ MyFraction new num: (num * f den) - (den * f num)
den: (den * f den)]

]

f5 := f1 + f2.

f6 := f1 - f2.

f5 printOn: Transcript.

f6 printOn: Transcript.

120/24

0/24

Assignment Project Exam Help

*** and /**

Add WeChat powcoder

MyFraction extend [

* f [^ MyFraction new num: (num * f num)
den: (den * f den)]

/ f [^ MyFraction new num: (num / f den)
den: (den * f num)]

]

f7 := f1 * f2.

f8 := f1 / f2.

f7 printOn: Transcript.

f8 printOn: Transcript.

150/24

60/60

Assignment Project Exam Help

= and ~=

Add WeChat powcoder

MyFraction extend [

= f [^ (num * f den) = (den * f num)]

]

Assignment Project Exam Help

<https://powcoder.com>

f1 = f2 printOn: Transcript. Transcript cr.

f2 = f3 printOn: Transcript. Transcript cr.

f3 = f4 printOn: Transcript. Transcript cr.

true
false
true

f1 ~= f2 printOn: Transcript. Transcript cr.

f2 ~= f3 printOn: Transcript. Transcript cr.

f3 ~= f4 printOn: Transcript. Transcript cr.

false
true
false

Assignment Project Exam Help

asFloat

Add WeChat powcoder

MyFraction extends **Assignment Project Exam Help**

```
asFloat [ ^ num asFloat / den asFloat ]  
]
```

<https://powcoder.com>

Add WeChat powcoder

f1 asFloat printOn: Transcript. Transcript cr.

f3 asFloat printOn: Transcript. Transcript cr.

f5 asFloat printOn: Transcript. Transcript cr.

f7 asFloat printOn: Transcript. Transcript cr.

2.5

-2.5

5.0

6.25

Assignment Project Exam Help

reduce

Add WeChat powcoder

MyFraction extend [

reduce [|g| Assignment Project Exam Help

g := num gcd: den.

den < 0 ifTrue: [g := 0 - g].

num := num / g Add WeChat powcoder

den := den / g]

]

f1 reduce. f2 reduce. f3 reduce. f4 reduce.

f5 reduce. f6 reduce. f7 reduce. f8 reduce.

Assignment Project Exam Help

reduce

Add WeChat powcoder

f1 printOn: Transcript.

f2 printOn: Transcript.

f3 printOn: Transcript.

f4 printOn: Transcript.

f5 printOn: Transcript.

f6 printOn: Transcript.

f7 printOn: Transcript.

f8 printOn: Transcript.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

5/2
5/2
-5/2
-5/2

5/1
0/1
25/4
1/1

Assignment Project Exam Help

Always reduce

Add WeChat powcoder

```
MyFraction extend [  
  num: n den: d [ num := n. den := d. self reduce ]  
]
```

Assignment Project Exam Help

```
f1 := MyFraction new num: 10 den: 4.
```

<https://powcoder.com>

```
f2 := MyFraction new num: 15 den: 6.
```

Add WeChat powcoder

```
f3 := MyFraction new num: -10 den: 4.
```

```
f4 := MyFraction new num: 15 den: -6.
```

```
f1 printOn: Transcript.
```

```
f2 printOn: Transcript.
```

```
f3 printOn: Transcript.
```

```
f4 printOn: Transcript.
```

5/2

5/2

-5/2

-5/2

Assignment Project Exam Help

Always reduce

Add WeChat powcoder

f5 := f1 + f2. Assignment Project Exam Help

f6 := f1 - f2.

<https://powcoder.com>

f7 := f1 * f2.

f8 := f1 / f2.

Add WeChat powcoder

f5 printOn: Transcript.

f6 printOn: Transcript.

f7 printOn: Transcript.

f8 printOn: Transcript.

5/1

0/1

25/4

1/1

Assignment Project Exam Help

Better = and ~=

Add WeChat powcoder

MyFraction extend [

= f [^ (num = f num) & (den = f num)]
]

<https://powcoder.com>

f1 = f2 printOn: Transcript. Transcript cr.
f2 = f3 printOn: Transcript. Transcript cr.
f3 = f4 printOn: Transcript. Transcript cr.

true
false
true

f1 ~= f2 printOn: Transcript. Transcript cr.
f2 ~= f3 printOn: Transcript. Transcript cr.
f3 ~= f4 printOn: Transcript. Transcript cr.

false
true
false

Assignment Project Exam Help

OOP: inheritance and late method binding

Add WeChat powcoder

Object extend: #A.

A extend: #B.

B extend: #C

C extend: #D.

D extend: #E.

<https://powcoder.com>

Add WeChat powcoder

x := Array new: 5.

x at: 1 put: A new;

at: 2 put: B new;

at: 3 put: C new;

at: 4 put: D new;

at: 5 put: E new.

Assignment Project Exam Help

Object and self

Add WeChat powcoder

Object extend [
 m [^ 1]
 n [^ self m] <https://powcoder.com>
 output [Transcript display: self m; space;
 Add WeChat powcoder
 display: self n; tab]
]

Object new output.
Transcript cr.

1	1
---	---

Assignment Project Exam Help

Subclass A and super

Add WeChat powcoder

A extend [
p [^ self n]
q [^ super m] <https://powcoder.com>
output [super output.
Transcript display: self p; space;
display: self q; cr]
]

(x at: 1) output.

1	1	1	1
---	---	---	---

Assignment Project Exam Help

Subclasses B and C

Add WeChat powcoder

B extend [
 m [²]
]

(x at: 2) output.

Assignment Project Exam Help

<https://powcoder.com>

2	2	2	1
---	---	---	---

Add WeChat powcoder

C extend [
 p [^{super q}]
 q [^{super p}]
]

(x at: 3) output.

2	2	1	2
---	---	---	---

Assignment Project Exam Help

Subclasses D and E

Add WeChat powcoder

D extend [
m [³
]

Assignment Project Exam Help

<https://powcoder.com>

(x at: 4) output.

Add WeChat powcoder

E extend [
n [⁴
]

(x at: 5) output.

3	3	1	3
---	---	---	---

3	4	1	4
---	---	---	---

Assignment Project Exam Help

Iterators to:do: and do:

Add WeChat powcoder

1 to: 5 do: [:k | y := x at: k.
y output].

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

1	1	1	1
2	2	2	1
2	2	1	2
3	3	1	3
3	4	1	4

x do: [:each | each output]

1	1	1	1
2	2	2	1
2	2	1	2
3	3	1	3
3	4	1	4