Control Flow

if else **Statement**

if condition:

    indented statement(s)

indented statement(s) # execution continues

```
x = raw_input("enter number: ")
if x < 0:
    x = abs(x)
```

if else **Statement**

if condition:

    indented statement(s)

else:

    indented statement(s)

```
def max_two(a, b):
    if a < b:
        return b
    else:
        return a
```

```
>>> max(10, 15)
15
>>> max('Mozzarella', 'brie')
'brie'
```

if else Statement with Logical Operators

```
def test_increase(a, b, c):
    if a < b and b < c:
        return True
    else:
        return False
```

```
>>> test_increase(2, 6, 4)
False
>>> test_increase('BRIE', 'BRie', 'brie')
True
```

Nested *if–elif–else* Statement

```
if condition_1:
    <statements_block_A>
elif condition_2:
     <statements_block_B>
elif condition_3:
     pass
else:
     <statements_block_C>
<statements_block_D>
```

- when do we execute A and D?
- when do we execute B and D?
- when do we execute C?
- when do we execute D?

Nested *if-elif-else* Statement

if condition_1:

statement_1

elif condition_2:

statement_2

else:

default

```
def compute_grade(score):
    if score > 90:
        return ‘A’
    elif score > 80:
        return ‘B’:
    else:
        return ‘C’
```

```
>>> compute_grade(85)
‘B’
```

Python Loop Construct

- Python has two loop constructs:

```
for <iterator> in <iterable>:
    <statements_block>
```

```
>>> for cheese in [“Mozzarella”, “Brie”]
___     print cheese
```

```
while  <test>:
    <statements_block_1>
<statements_block_2>
```

```
>>> sum = 0; counter = 0
>>> while counter < 1000:
___         sum = sum + counter
___         counter = counter + 1
```

for **Loop with** *range*

```
# ratio of consecutive Fibonacci numbers
fib_numbers = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
for i in range(1, len(fib_numbers)):
    x = fib_numbers[i]
    y = fib_numbers[i–1]
    print (i, float(x)/y)
```

- how do we change the flow of execution?
  1) continue
  2) break

Python *continue* Statement

```
for <iterator> in <iterable>:
    <statements_block_1>
    if <test>:
        continue
    <statements_block_2>
<statements_block_3>
```

```
# ratio of consecutive odd Fibonacci numbers
fib_numbers = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
for i in range(1, len(fib_numbers)):
    x = fib_numbers[i]
    y = fib_numbers[i–1]
    if (x%2 ==0) or (y%2==0):
      continue
    print (i, float(x)/y)
```

break **Statement**

```
for <iterator> in <iterable>:
    <statements_block_1>
    if <test>:
        break
    <statements_block_2>
<statements_block_3>
```

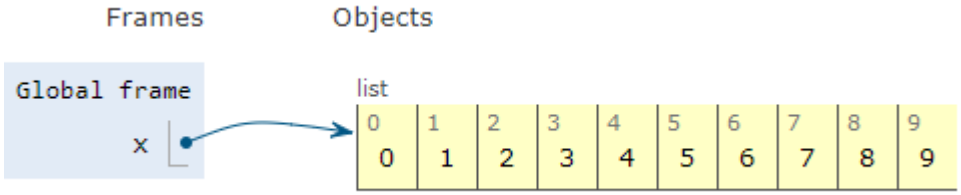
- if test is true, we exit loop and execute

```
# print first fib. Number > 25
fib_numbers = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
for i in range(1, len(fib_numbers)):
    x = fib_numbers[i]
    if (x > 25):
      break
    print ("first fibonacci number: ", x)
```

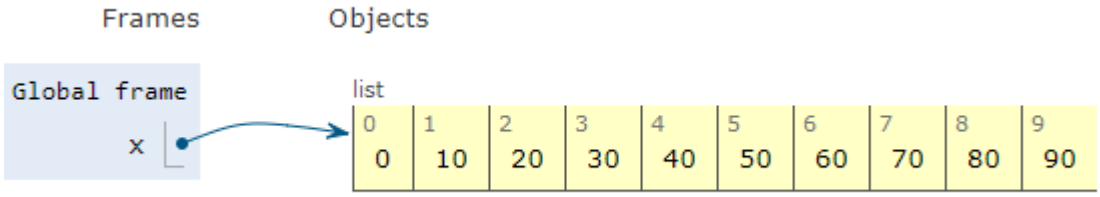
range() Examples

- format: *range*(*start*, *stop*[, *step*])
- default step is 1, default start 0

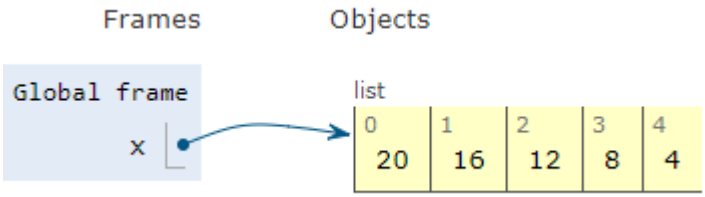
```
>>> x = range(0,10,1)
>>> x = range(10)          # shorthand
```




```
>>> x = range(0, 100, 10)   # count by 10
```

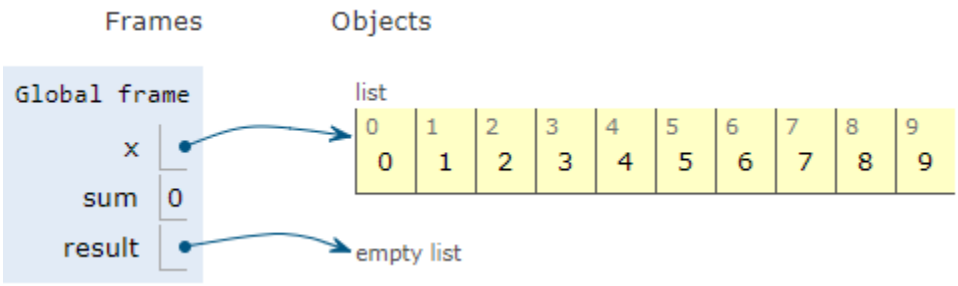



```
>>> x = range(20, 0, –4)   # backwards by 4
```

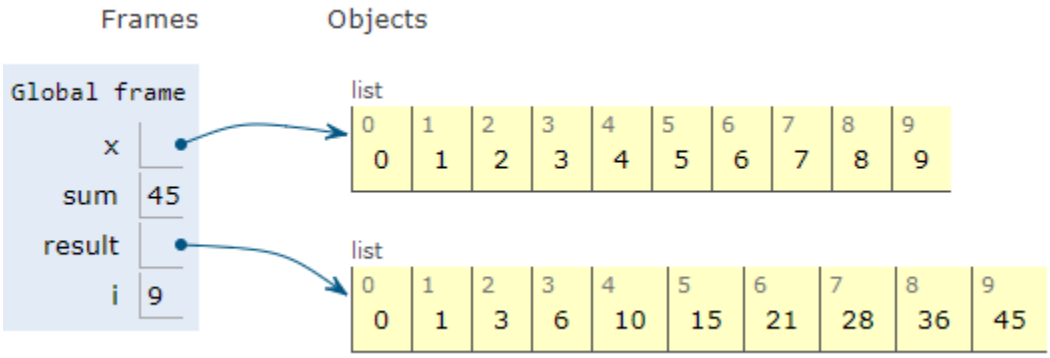


range() Example

```
>>> x = range(10)
>>> sum = 0
>>> list = []
```




```
>>> for i in x:
...         sum = sum + i
...         result.append(sum)
```




```
>>> print(result)
```

[0, 1, 3, 6, 10, 15, 21, 28, 36, 45]

Counting with range() Construct

- Python has two loop constructs:

```
for <iterator> in <iterable>:
    <statements_block>
```

```
>>> for cheese in [“Mozzarella”, “Brie”]
——      print cheese
```

```
while  <test>:
    <statements_block_1>
<statements_block_2>
```

```
>>> sum = 0; counter = 0
>>> while counter < 1000:
——          sum = sum + counter
——          counter = counter + 1
```

Review Problems

Interview Problem

- what is *pass* statement used for?

Interview Problem

- what are the two major loop statements?

Interview Problem

- explain the use of break and continue in looping

Interview Problem

- when would you use a *break* statement in a *for* loop?

Interview Problem

- what is the structure of a *for* loop?

Interview Problem

- what is the structure of a *while* loop?

Interview Problem

- under what circumstances would von use a *while* statement rather than *for*?

Interview Problem

- when would you use a *continue* statement in a *for* loop?

Interview Problem

- which statement of Python is used whenever a statement is required syntactically but the program needs no action?

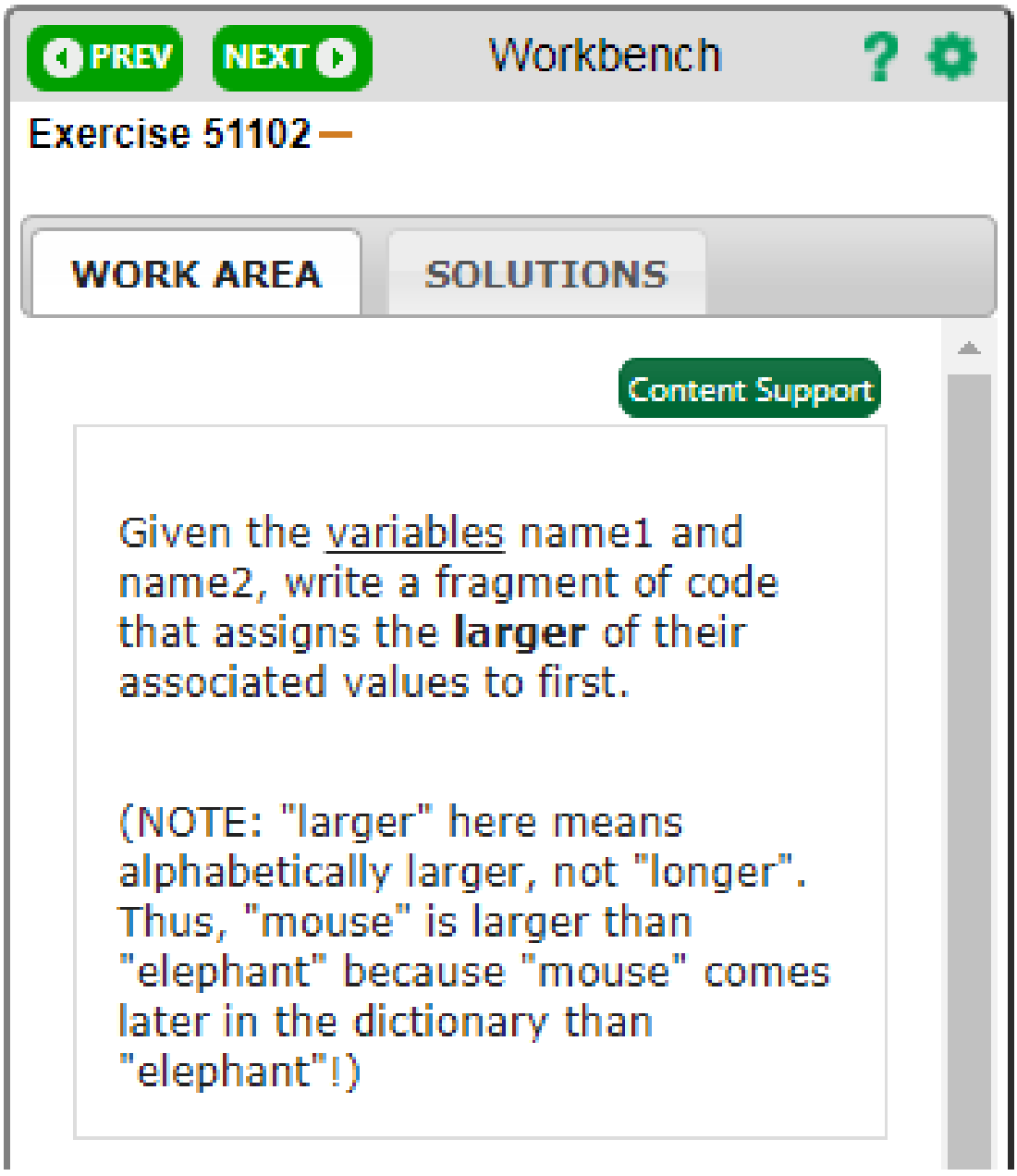
Interview Problem

- what are the optional statements to use inside a <try – except> block?

Interview Problem

- print sum of numbers from 1 to 100

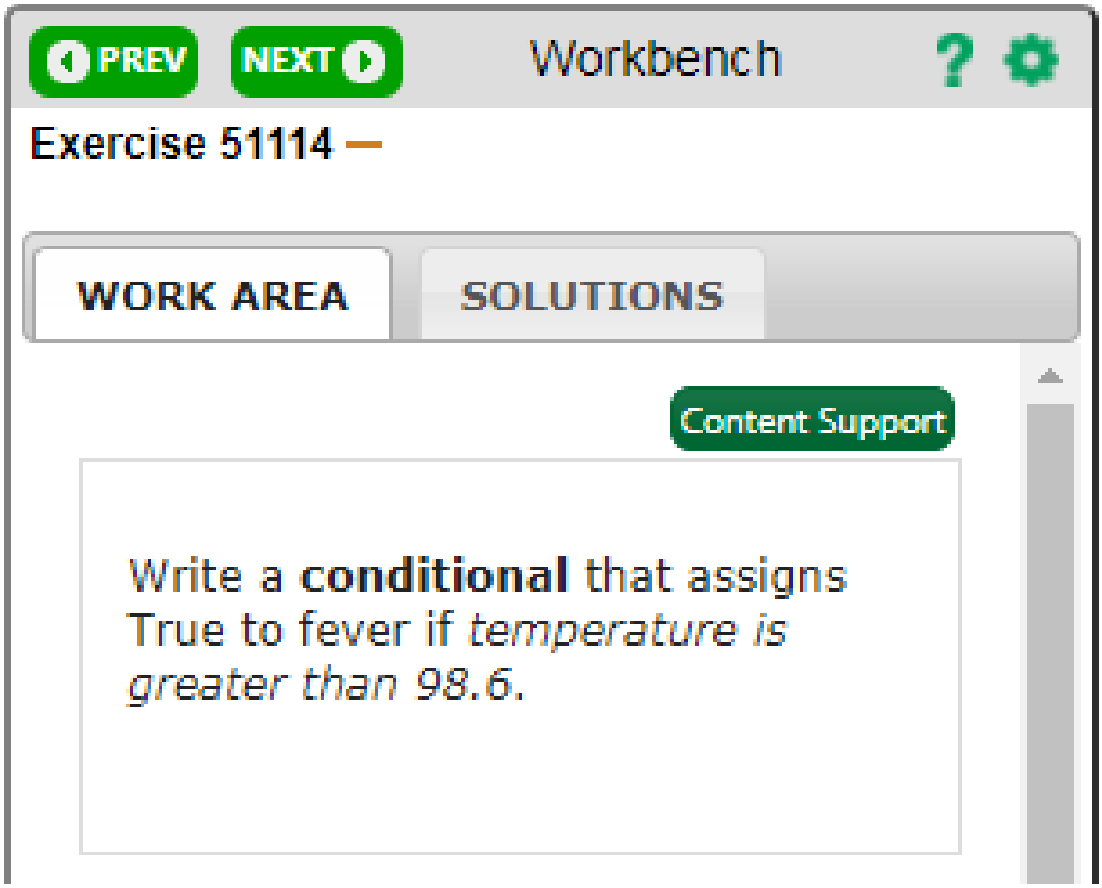
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

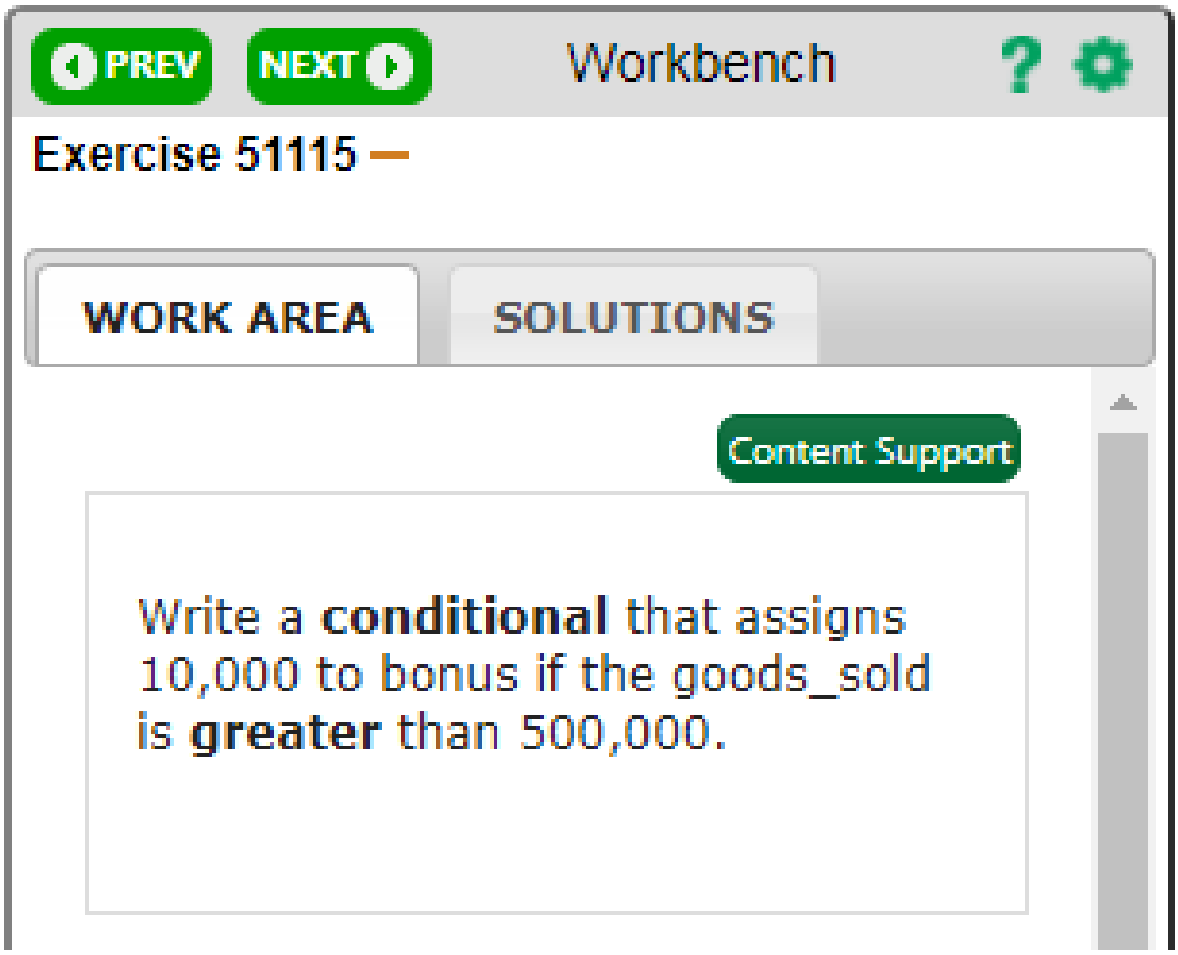
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

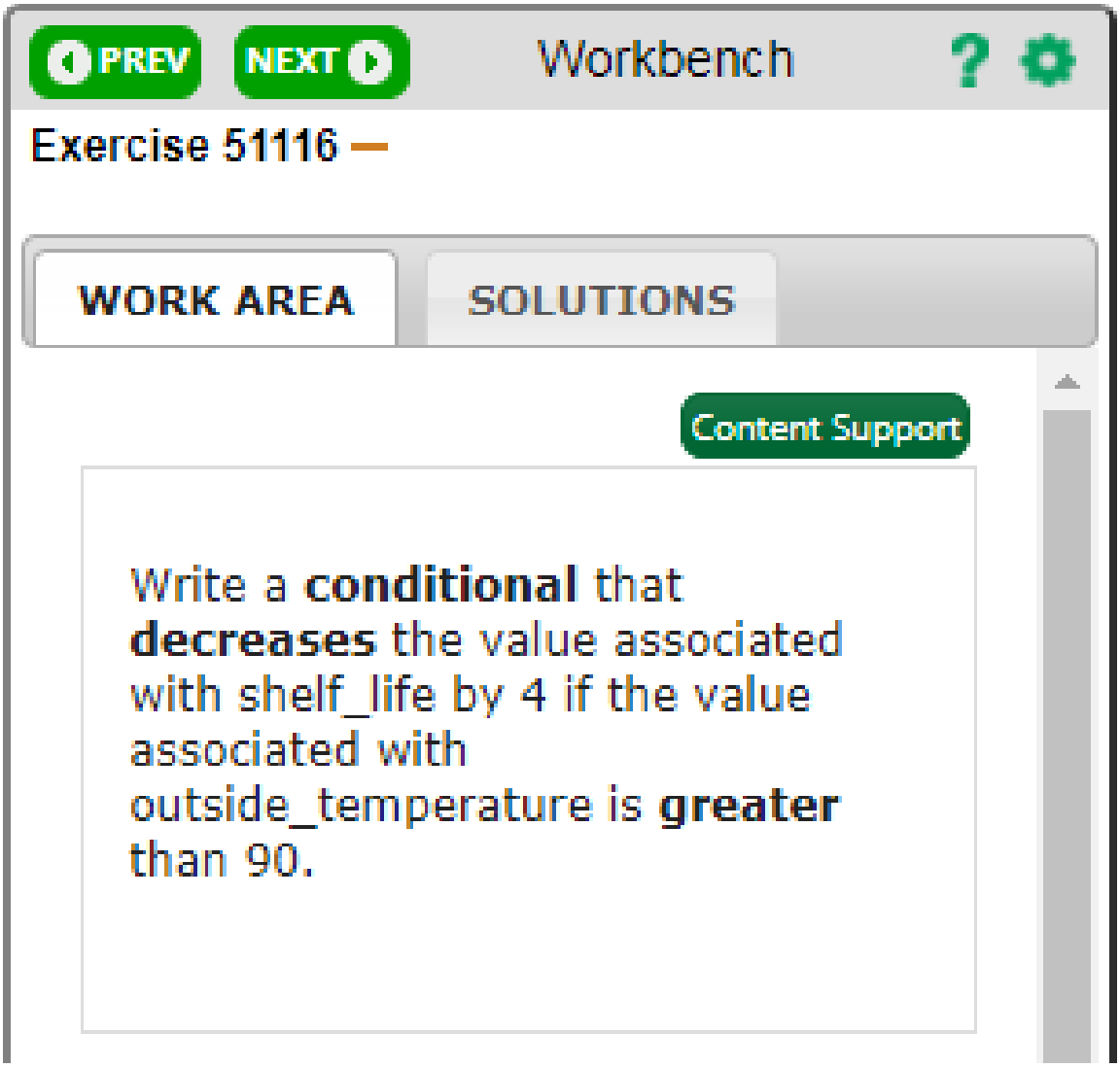
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

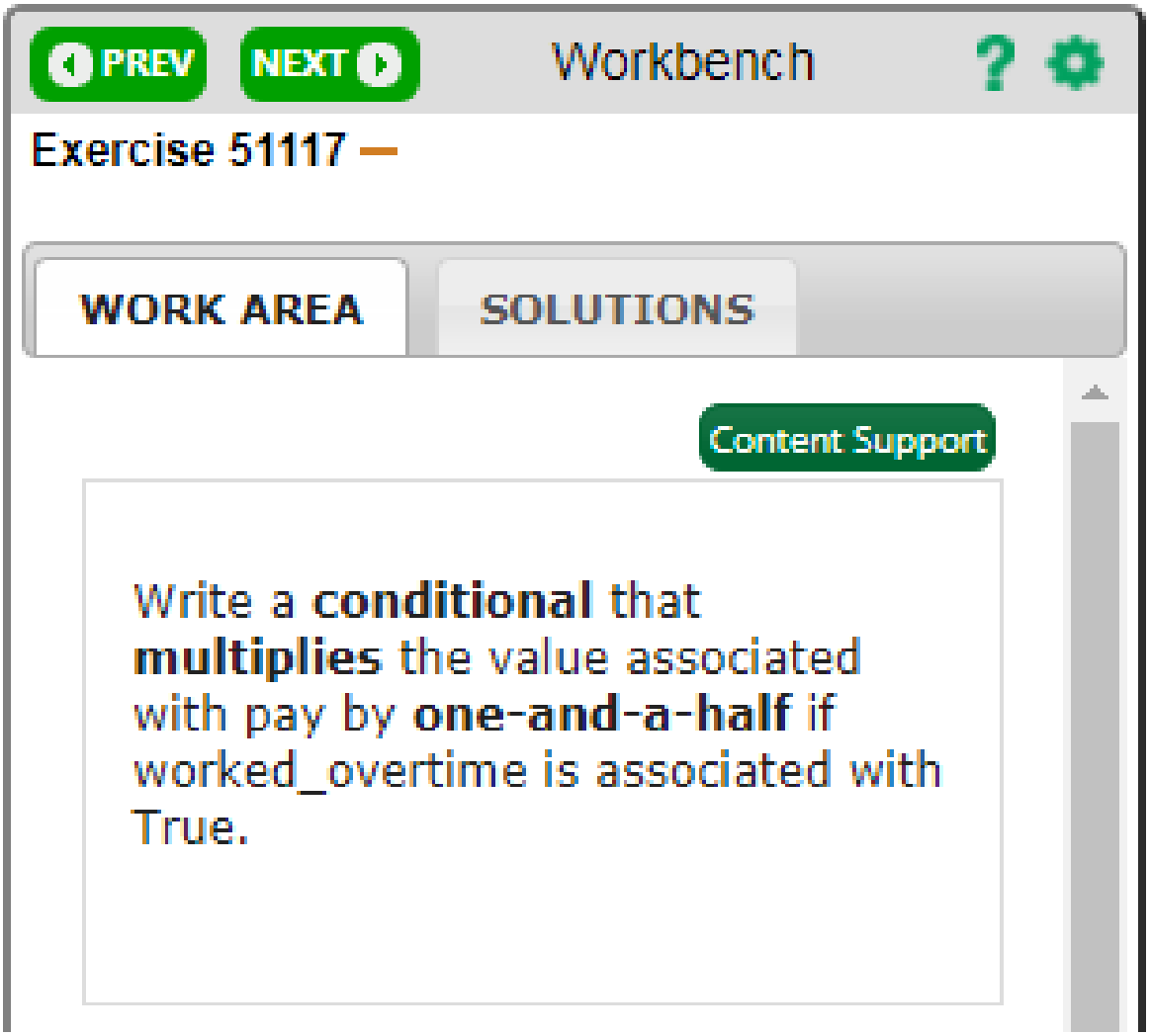
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

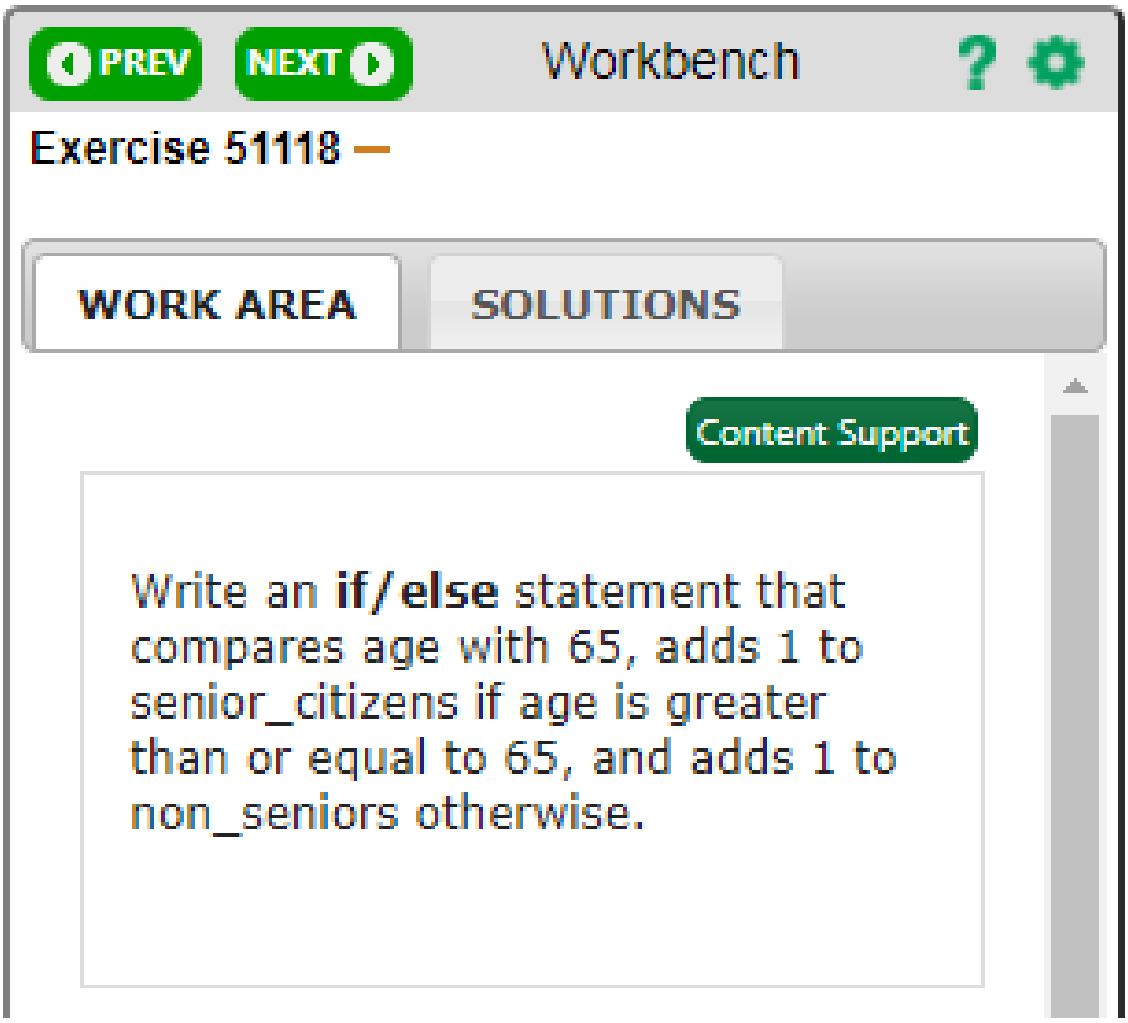
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

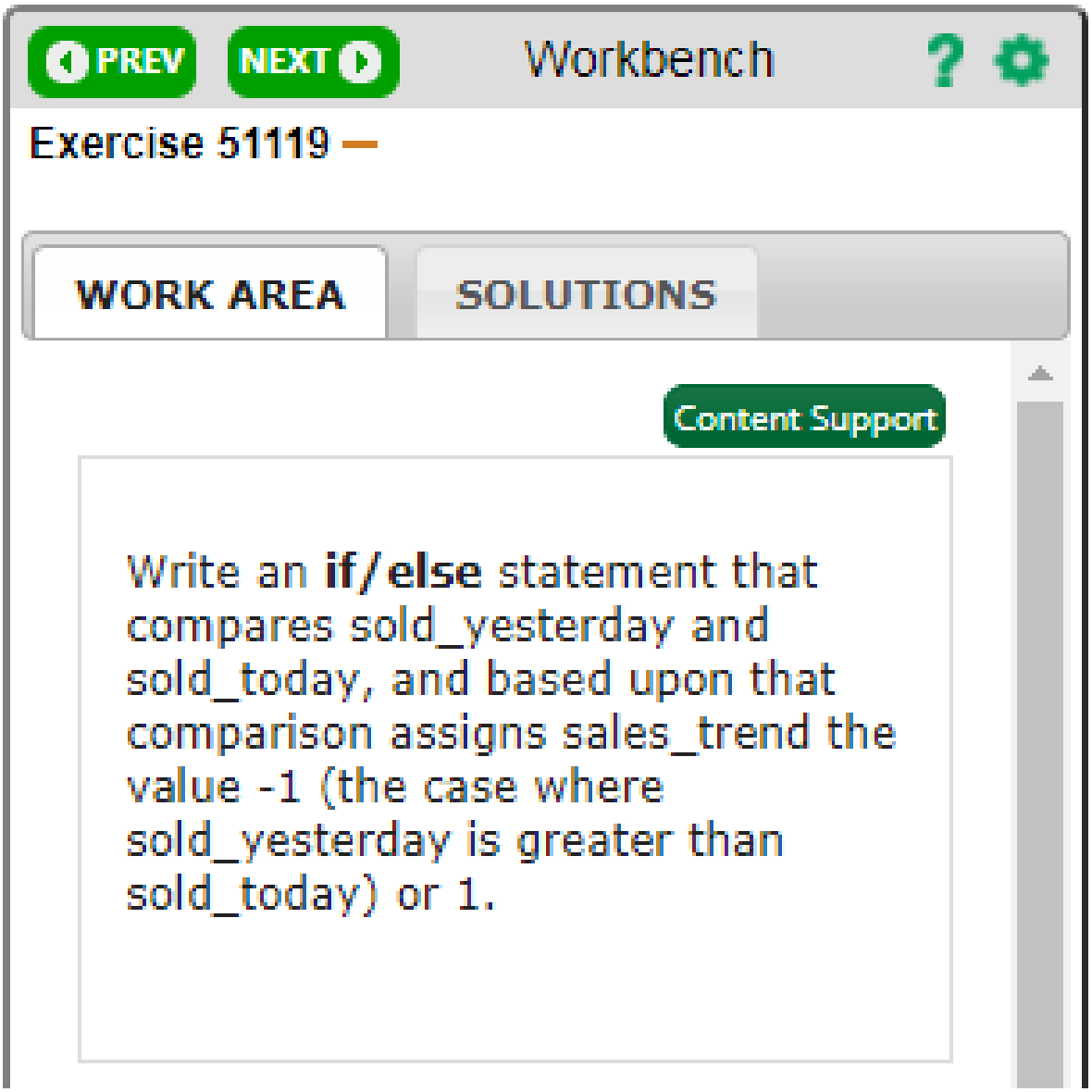
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

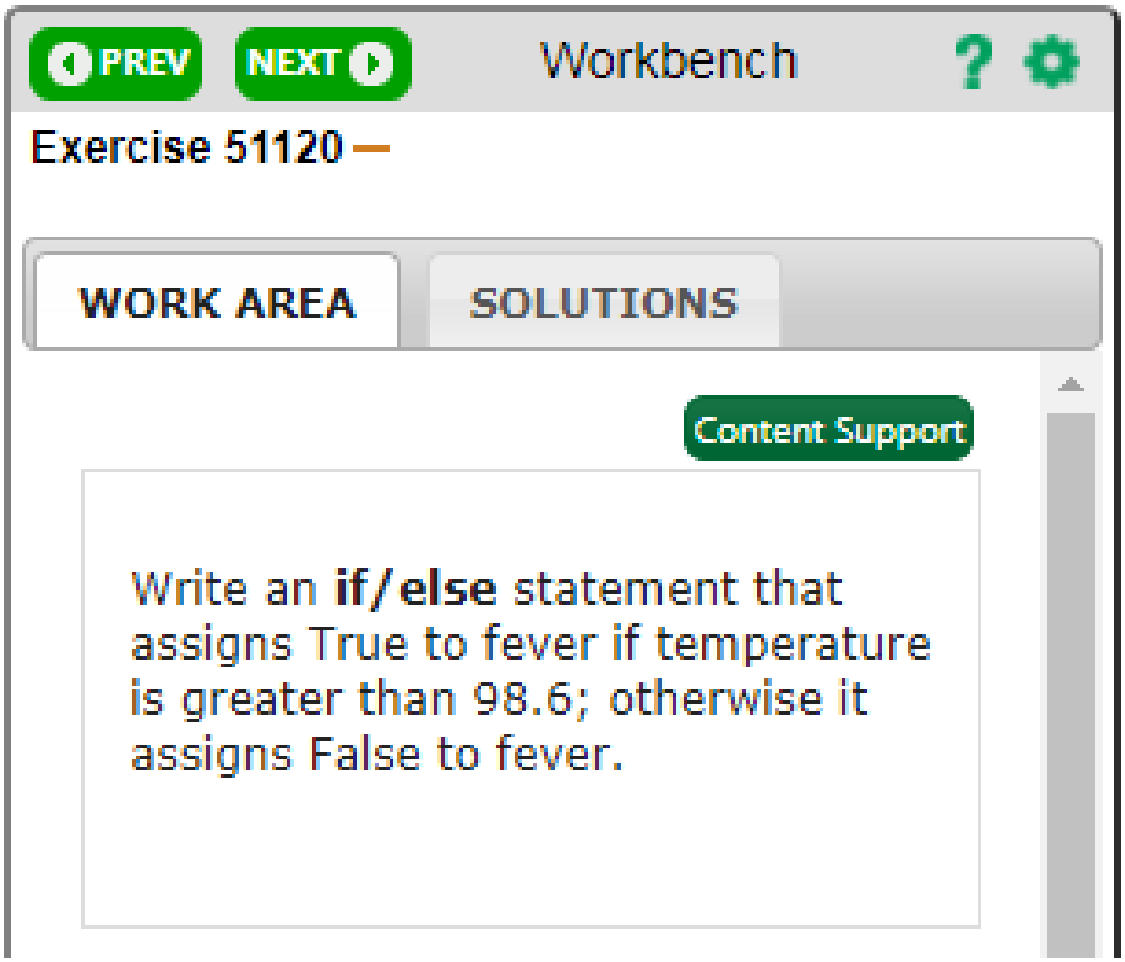
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

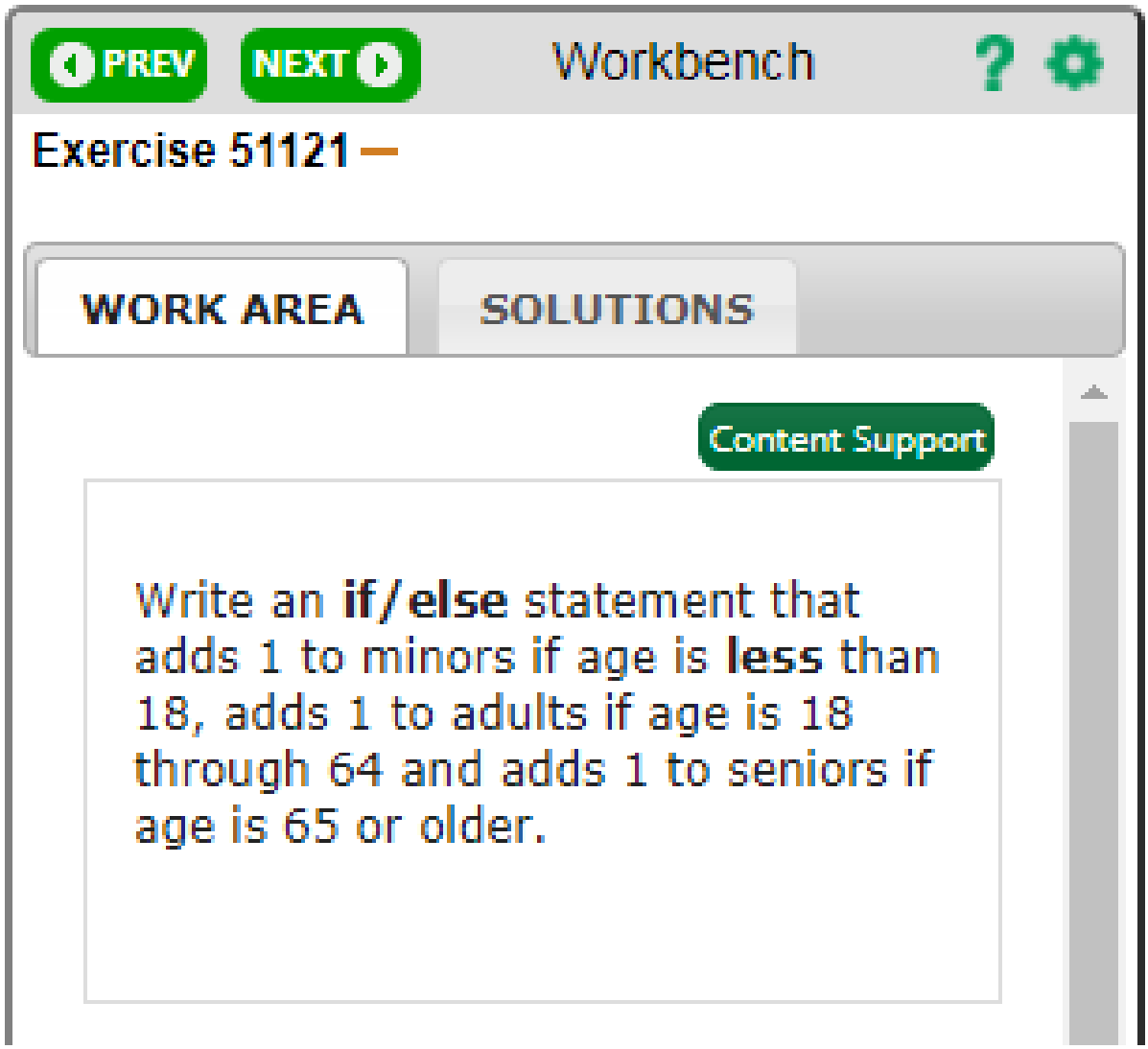
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

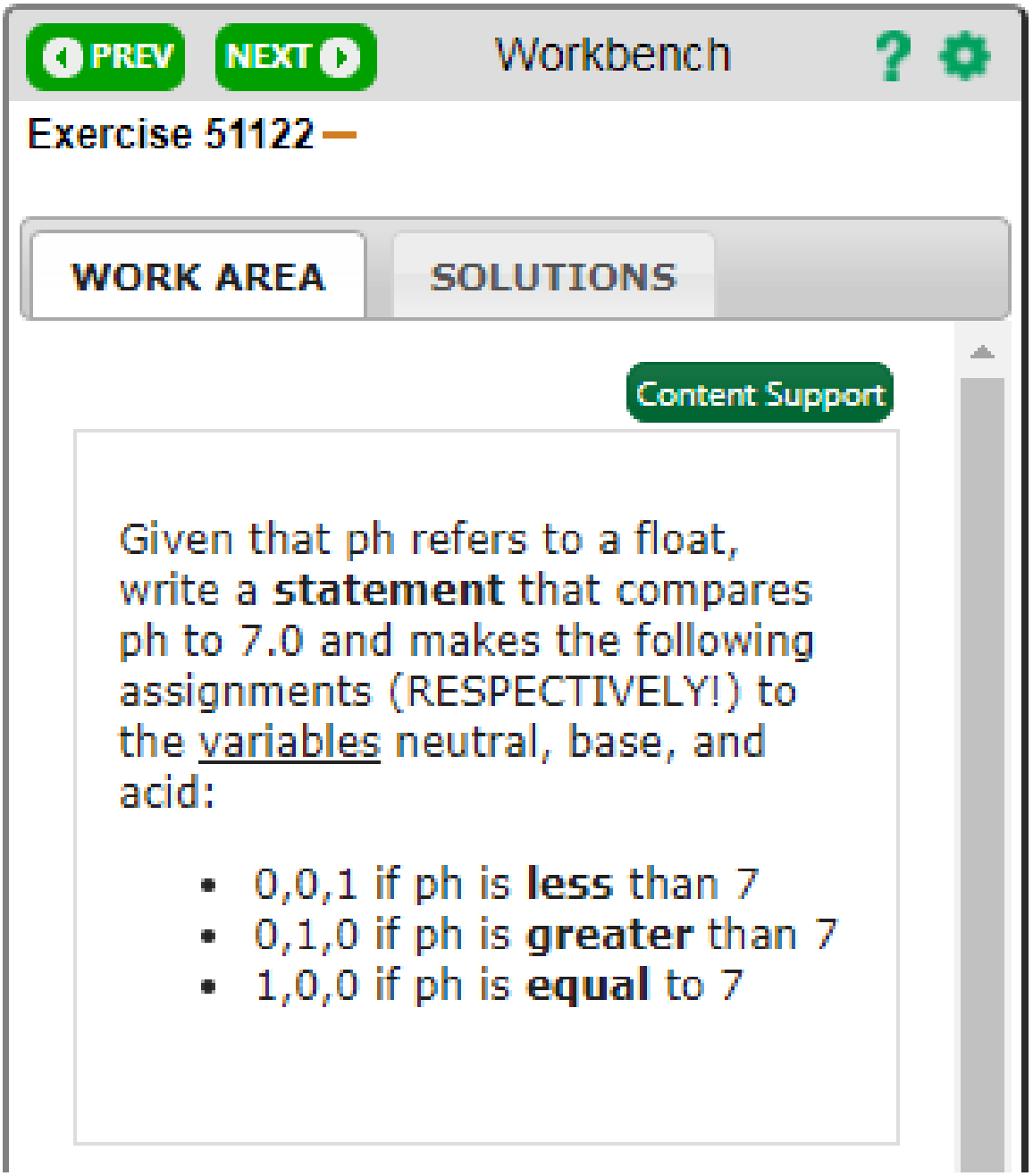
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

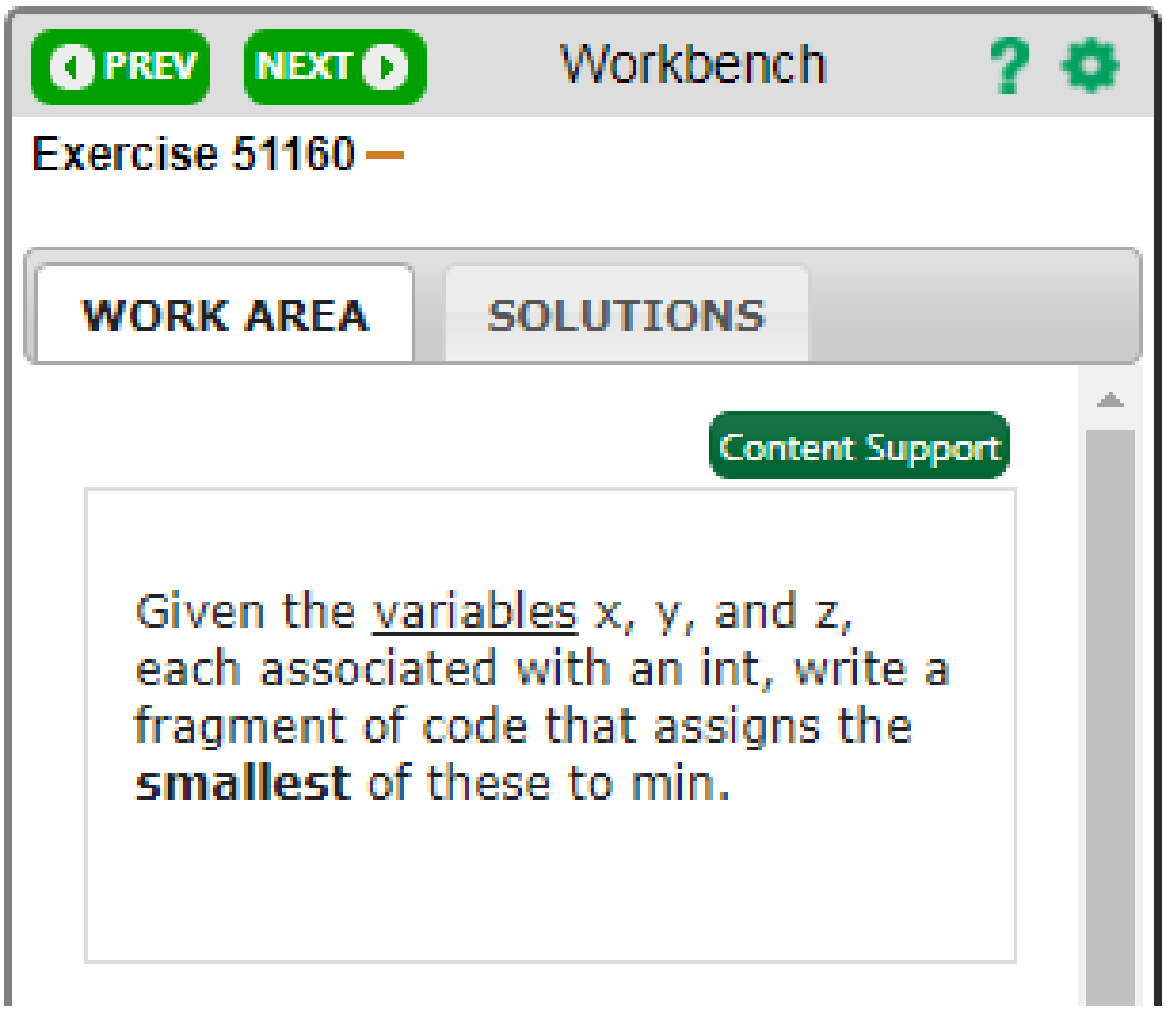
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

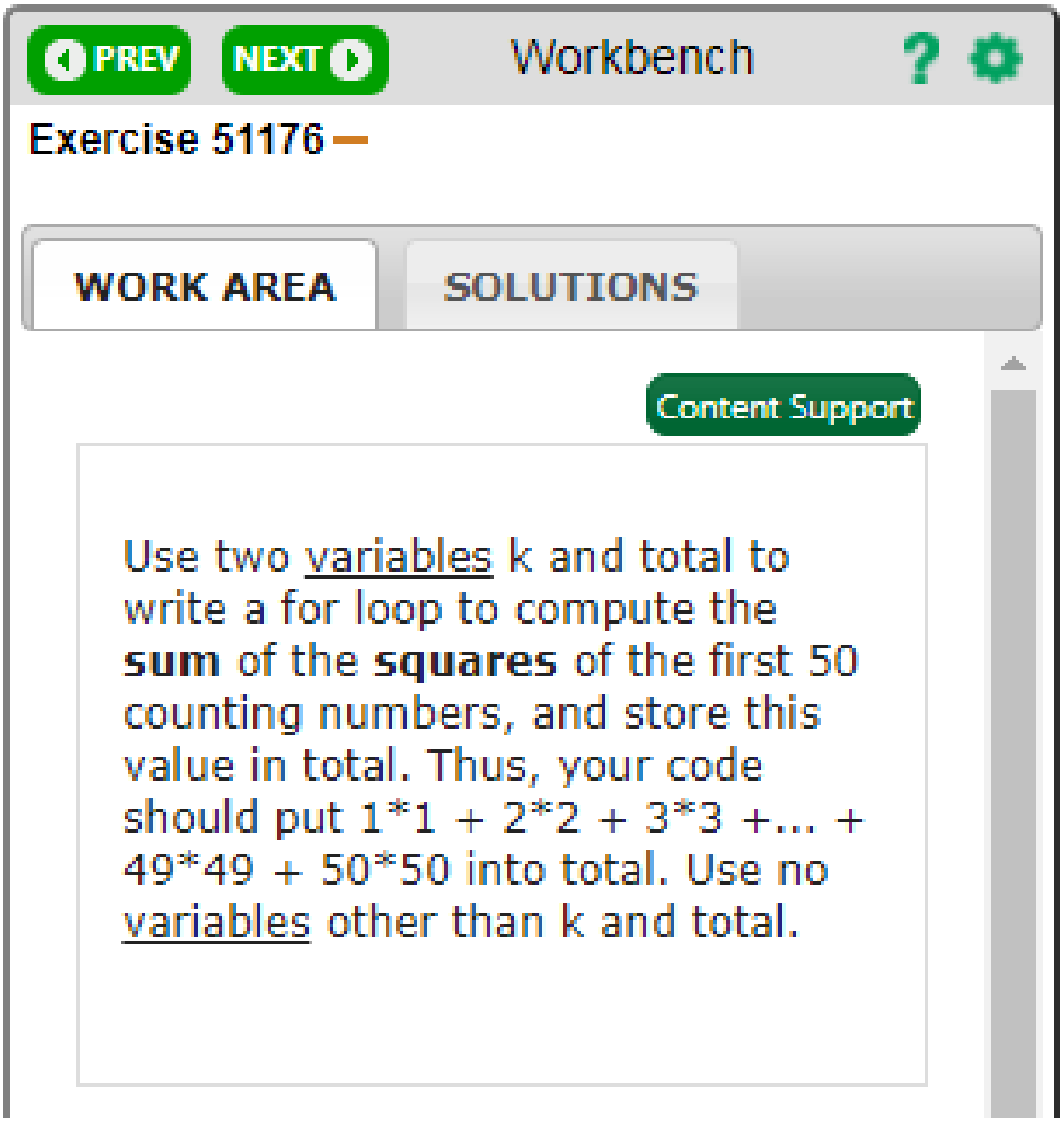
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

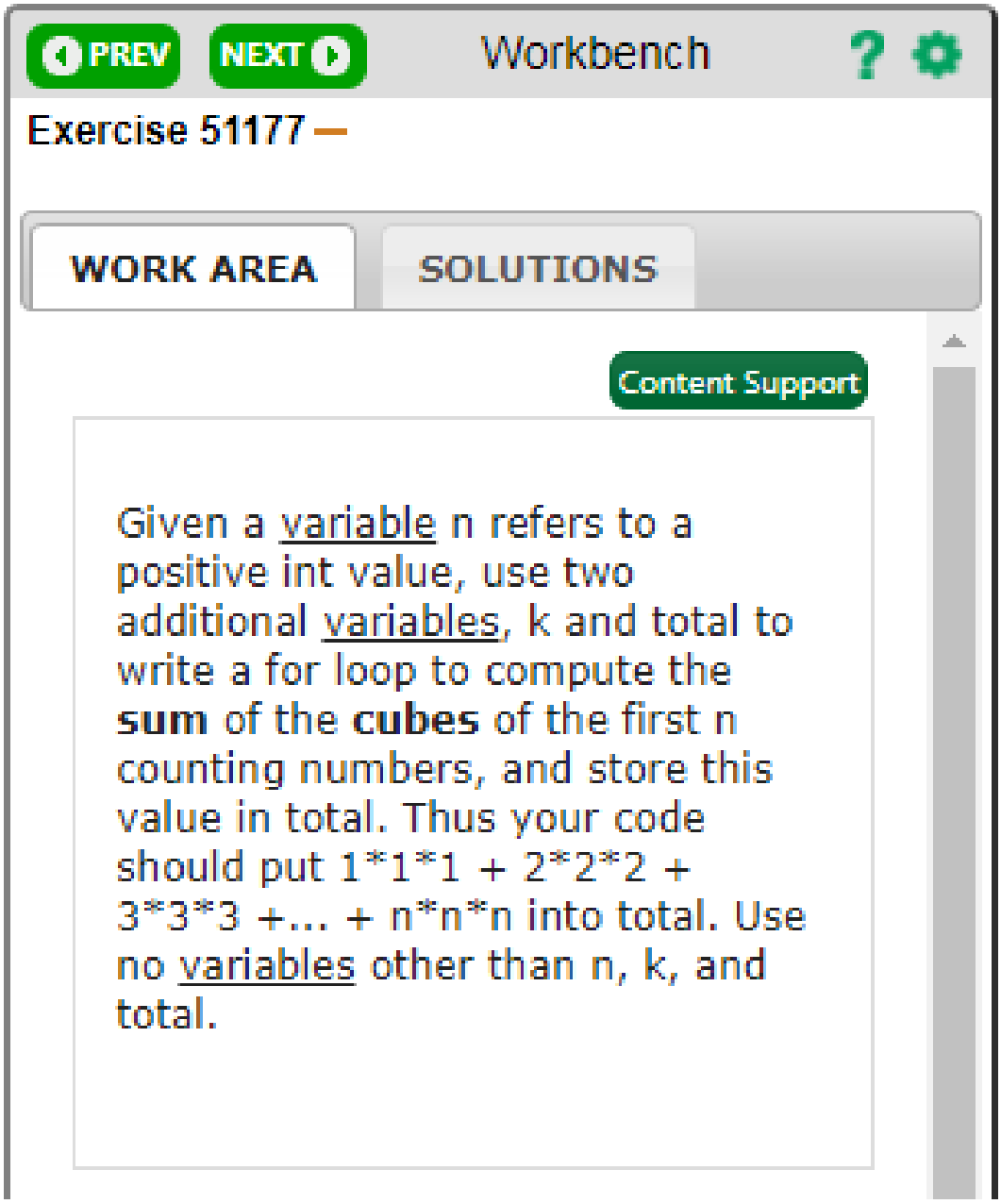
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

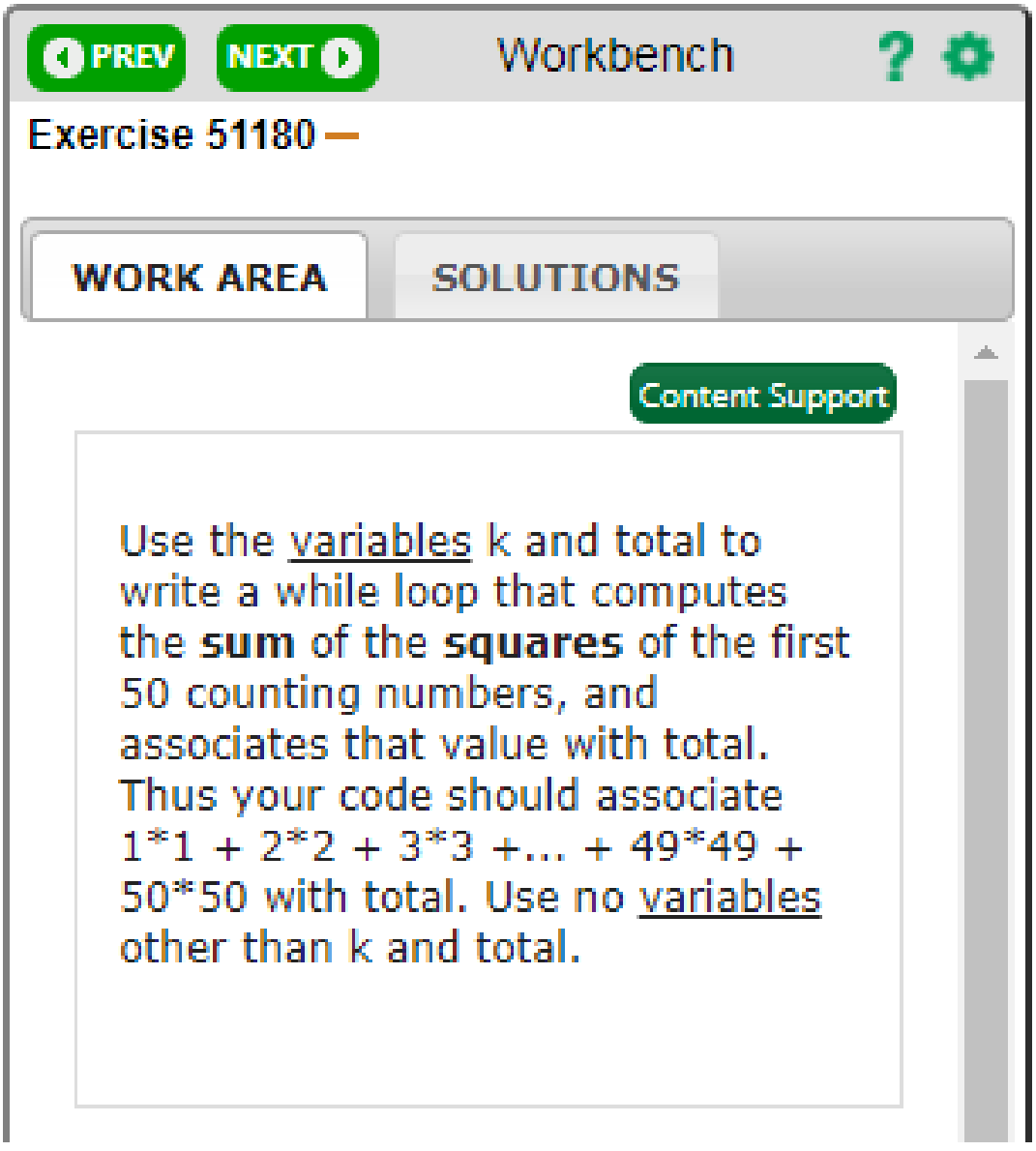
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

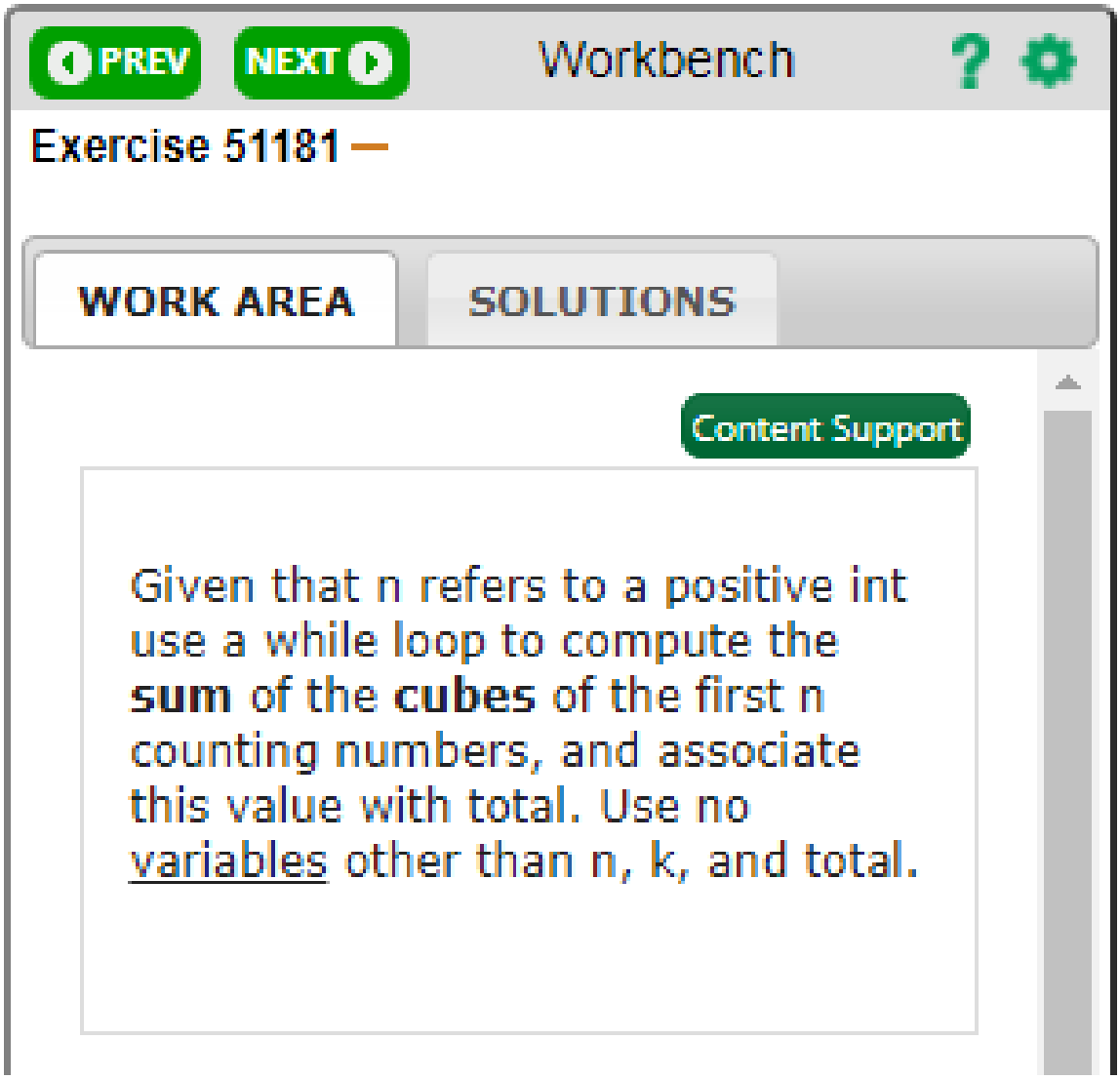
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

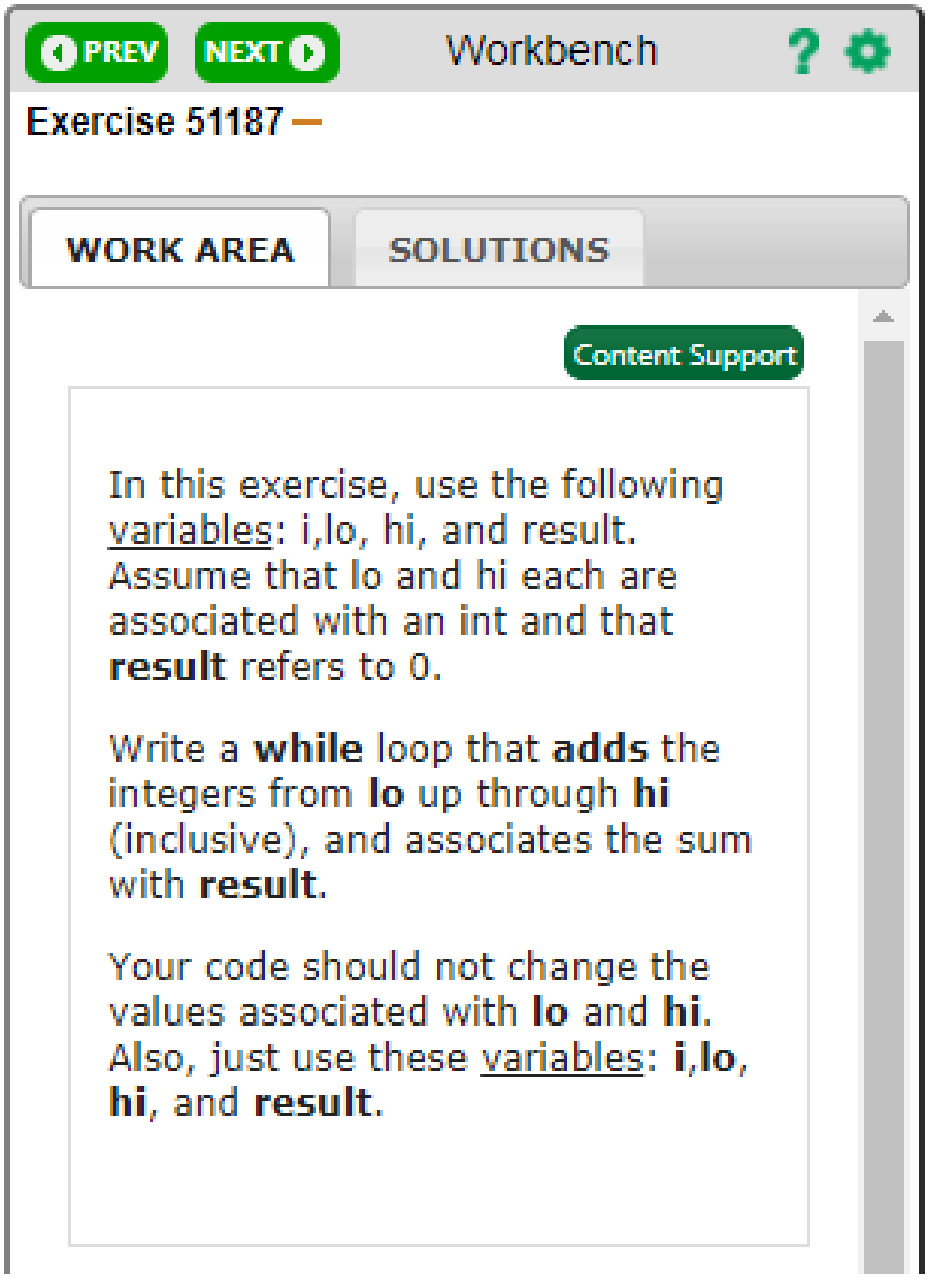
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

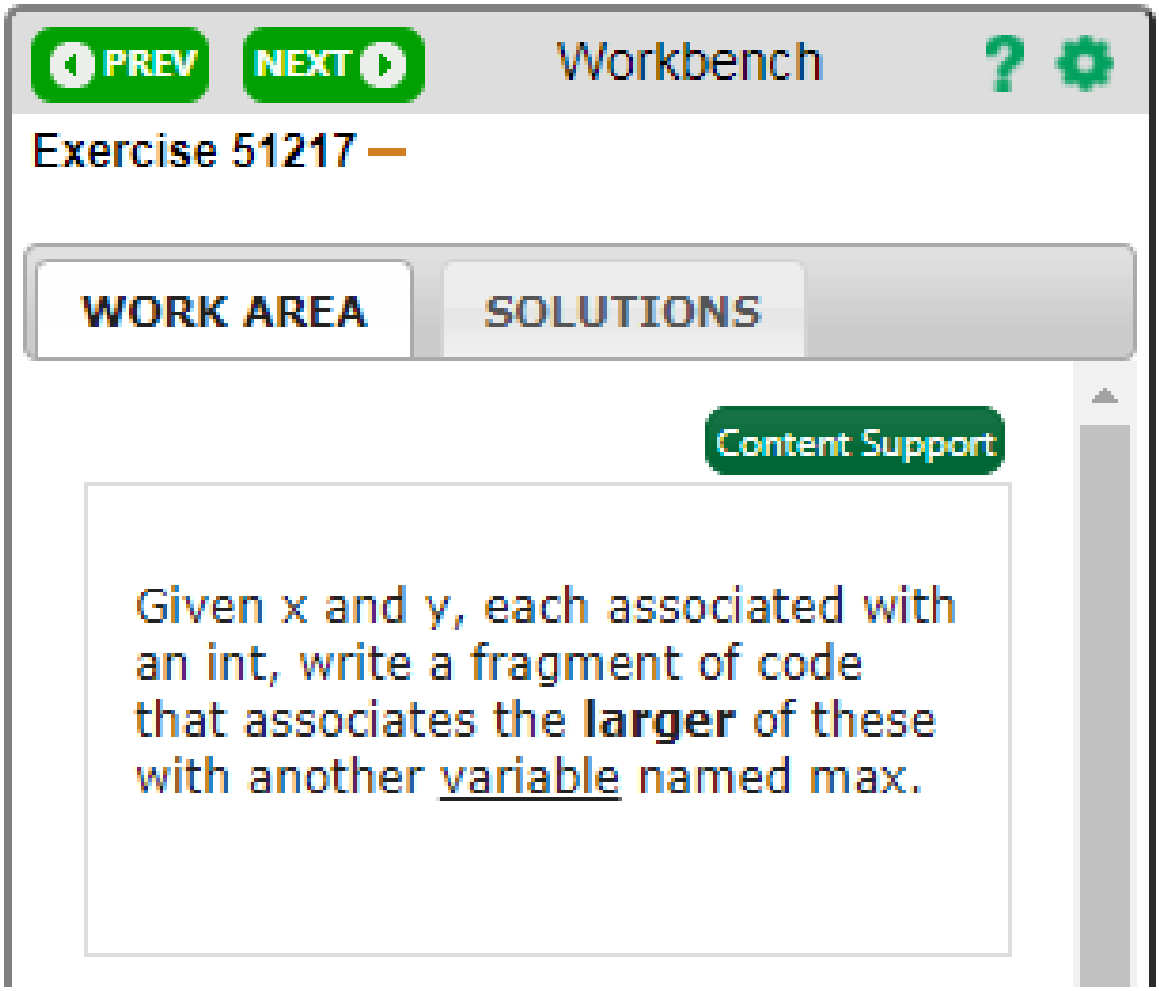
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

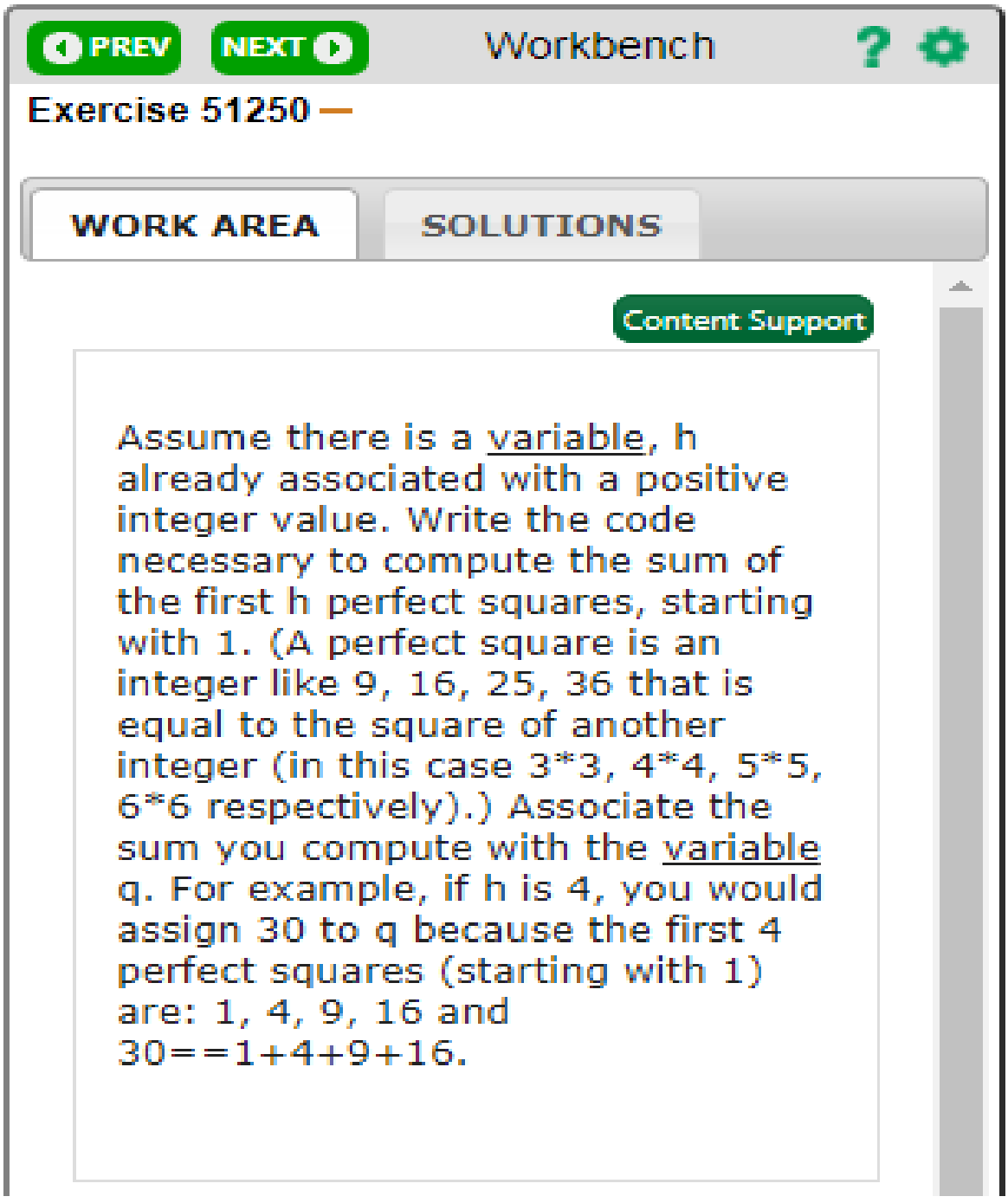
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

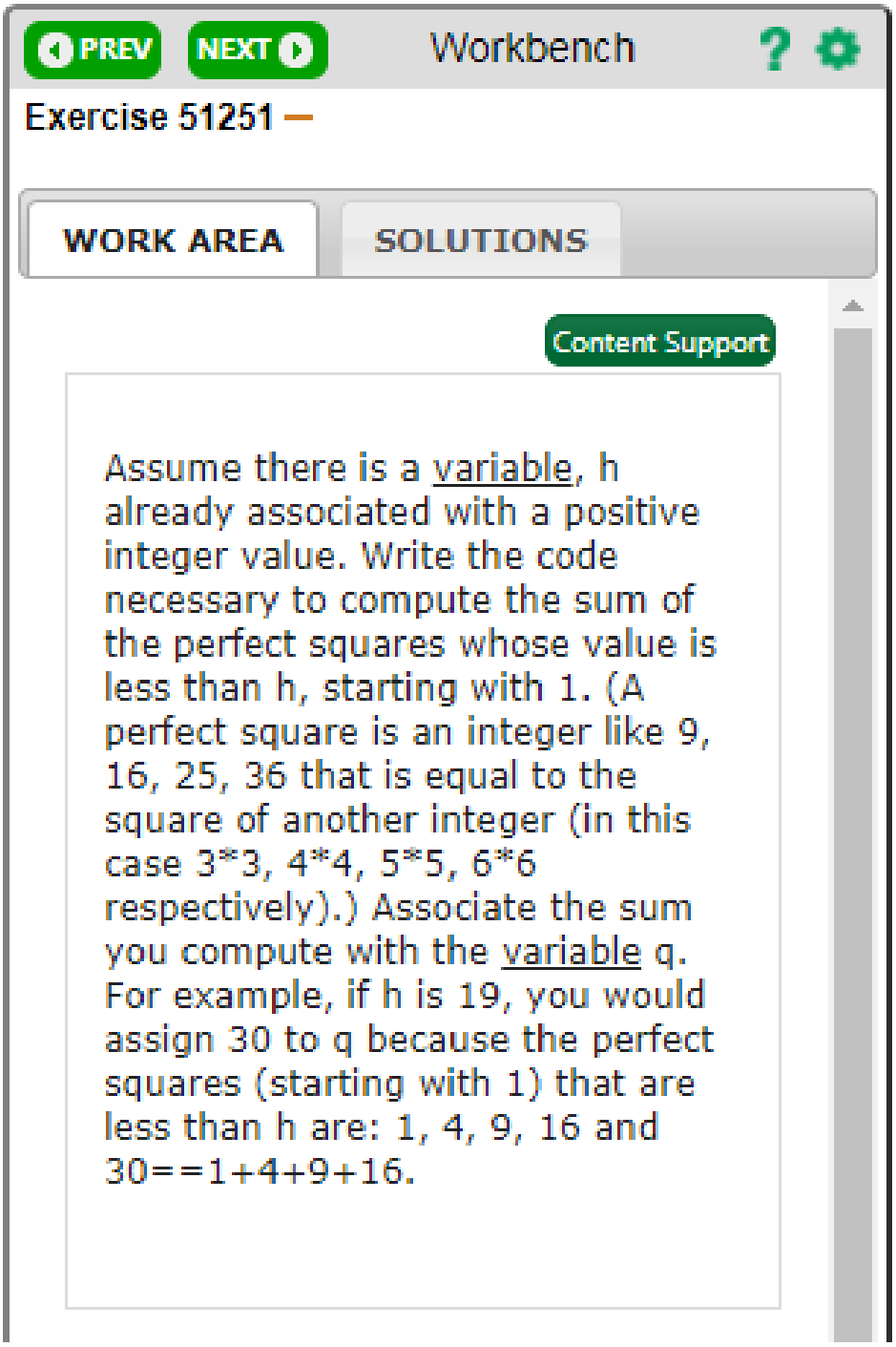
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

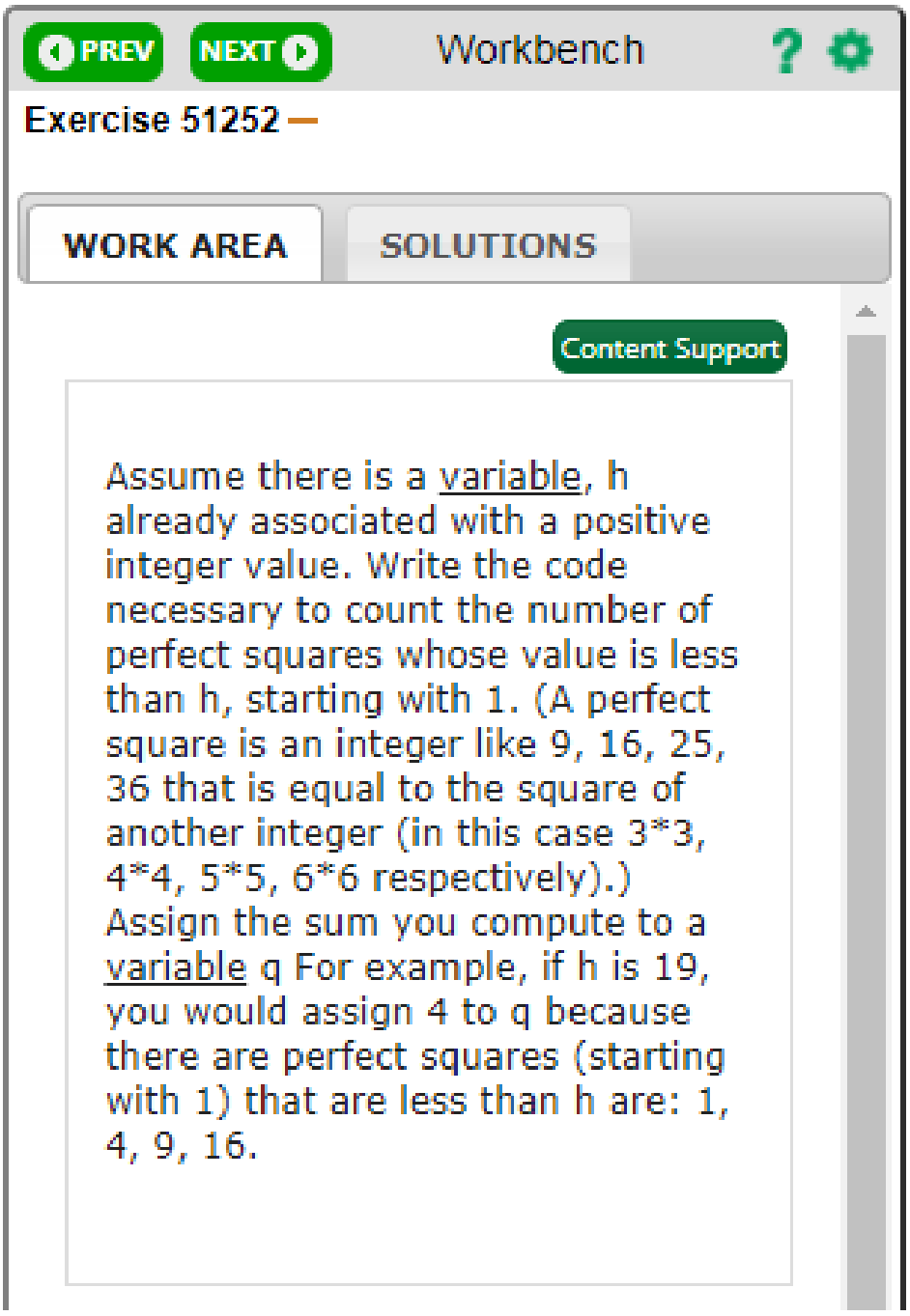
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

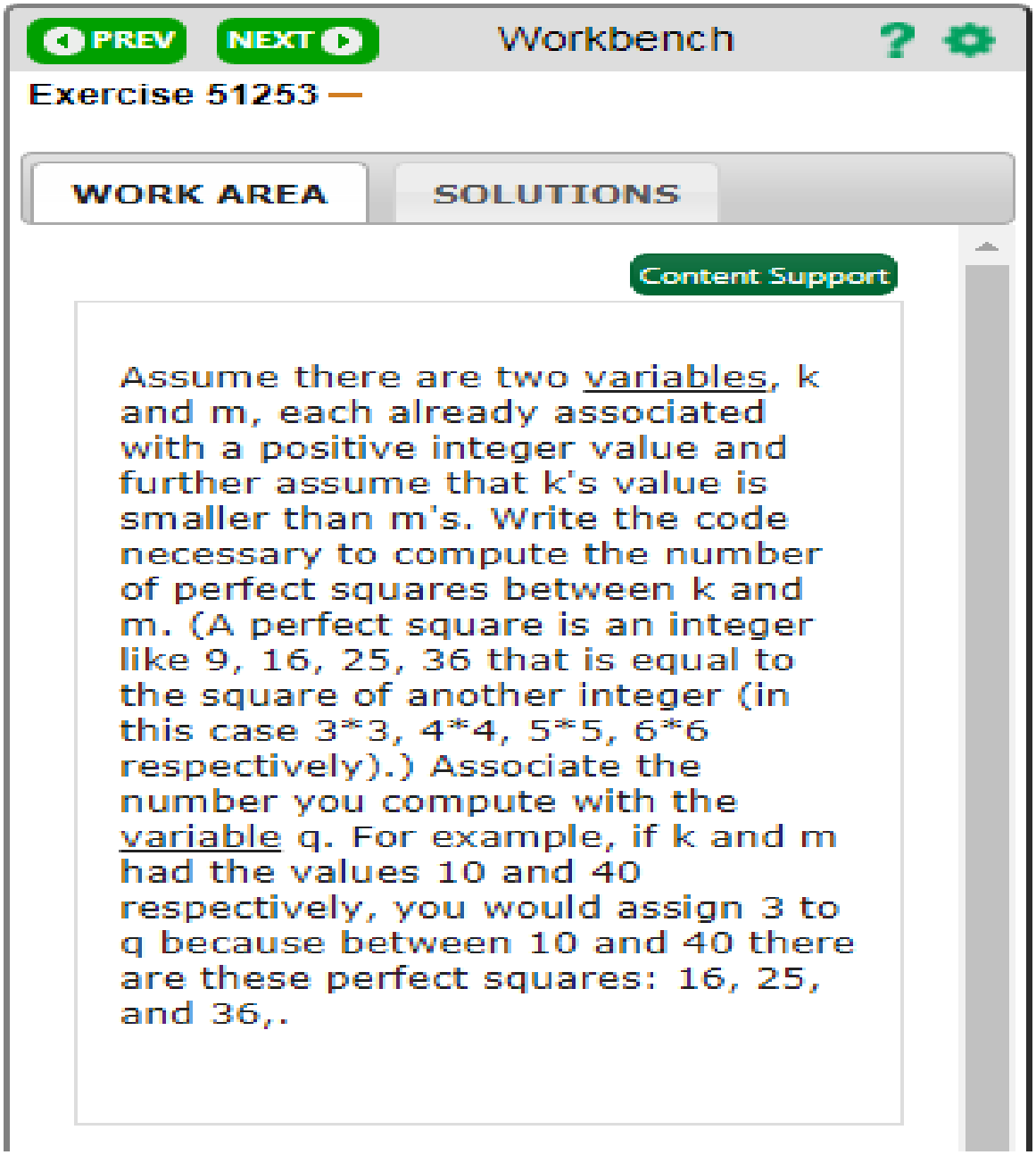
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

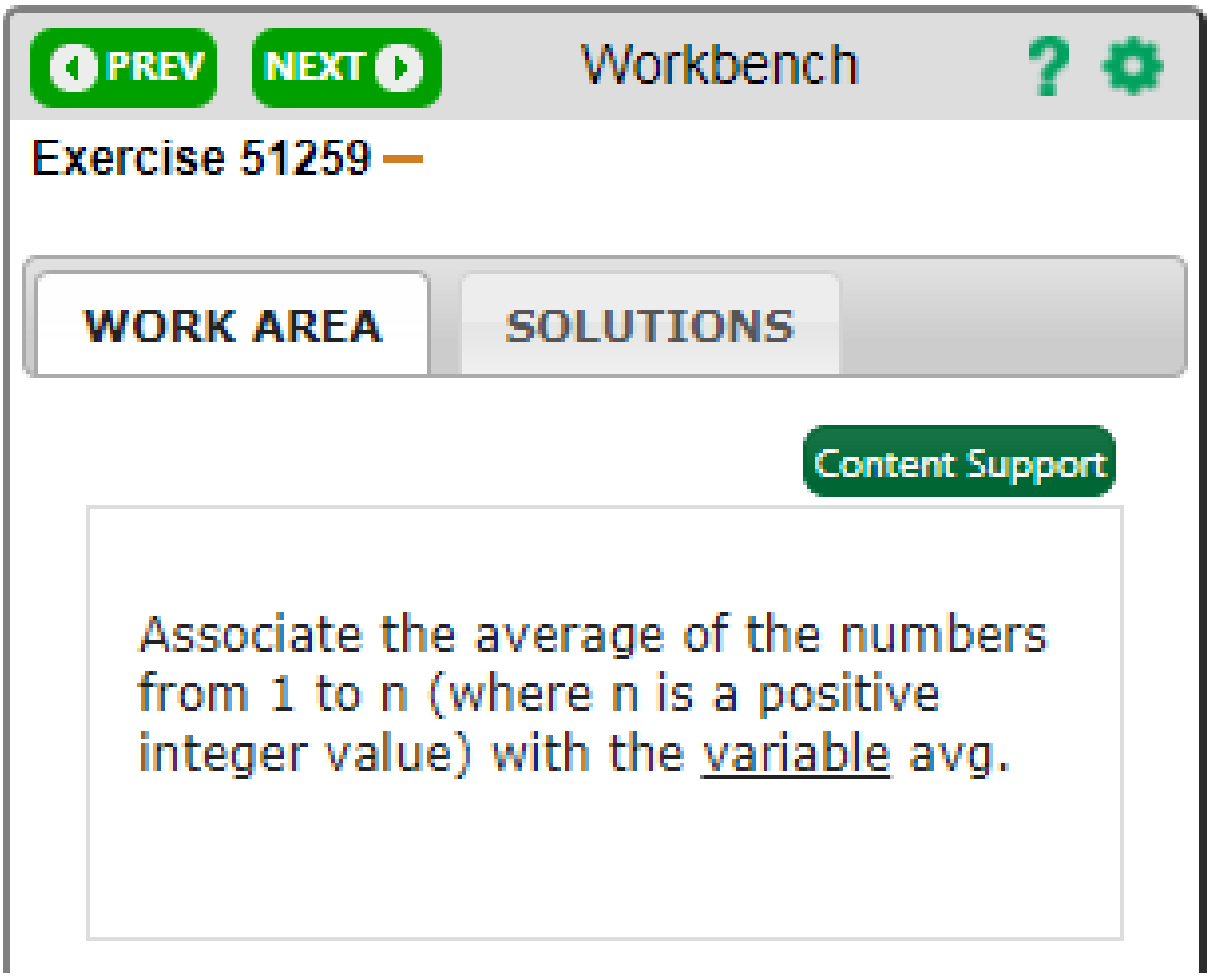
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

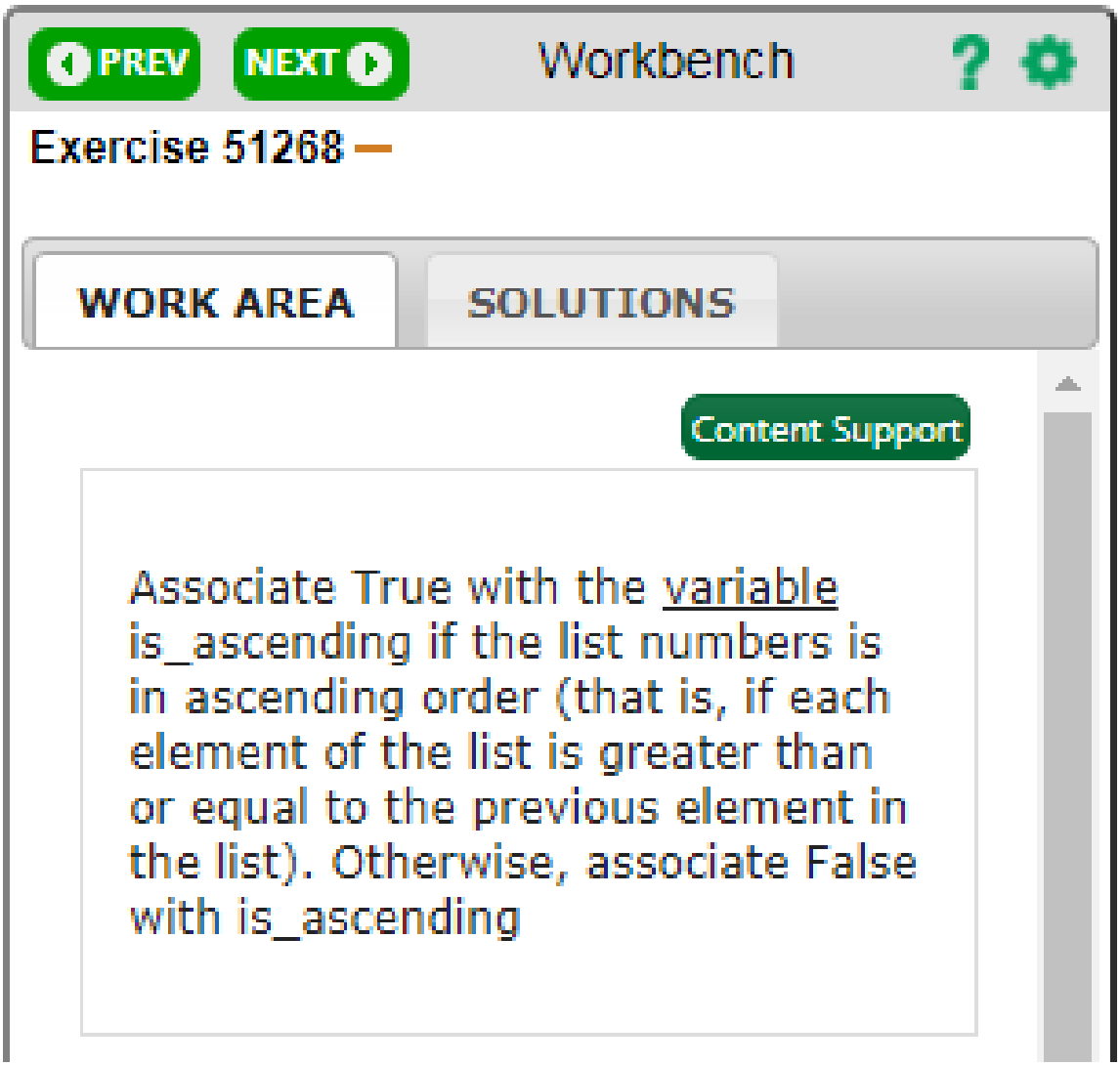
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

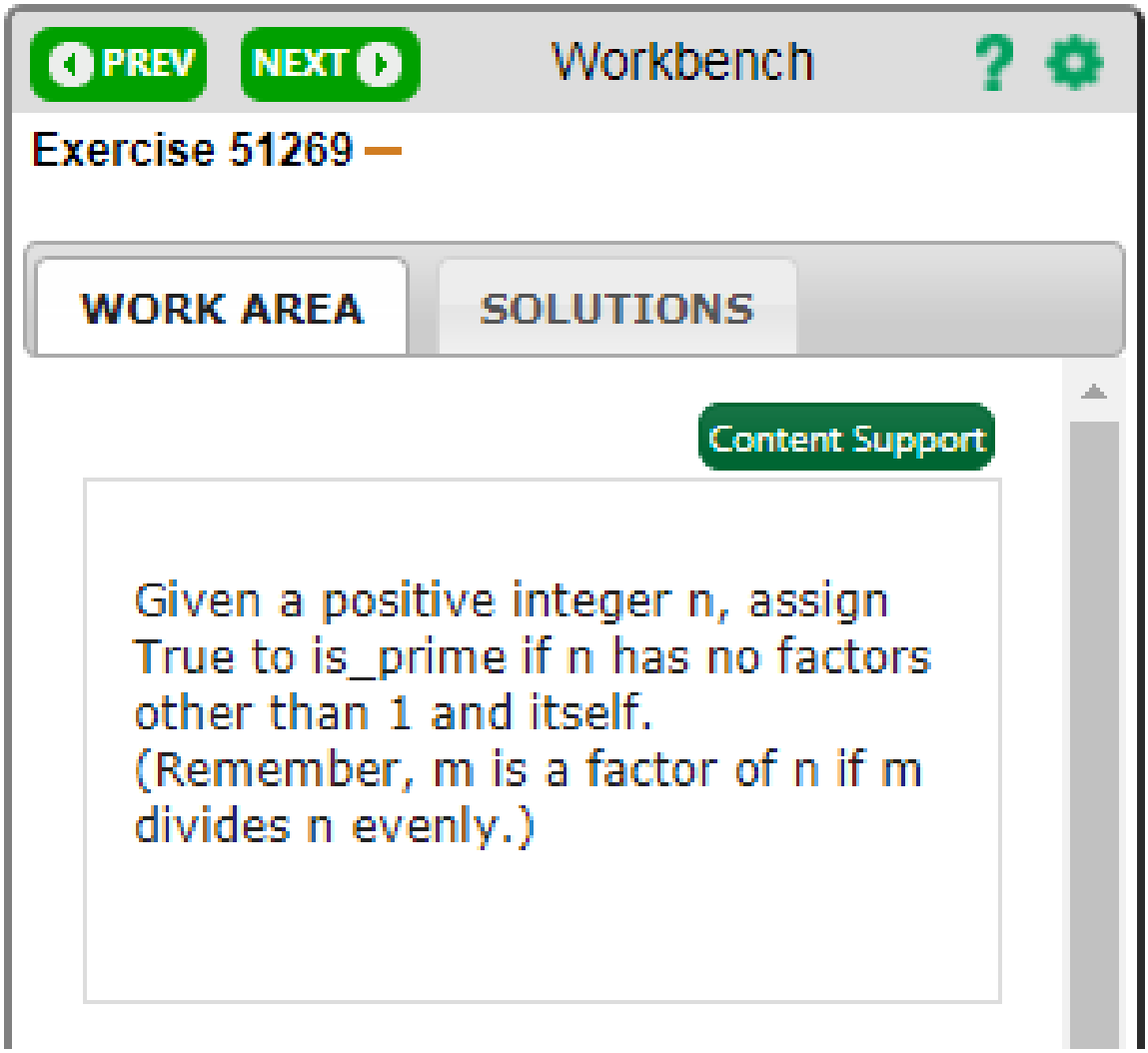
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

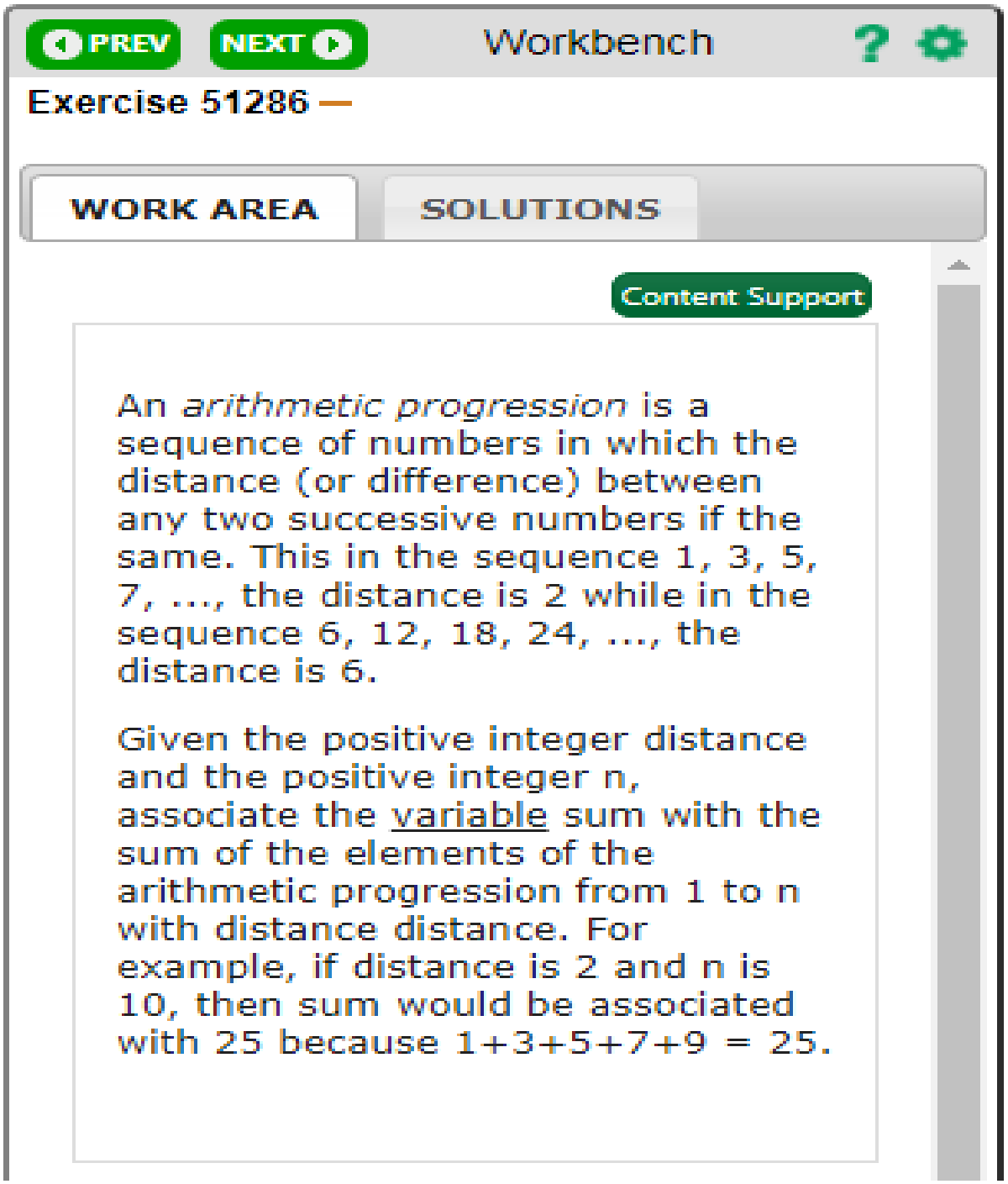
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

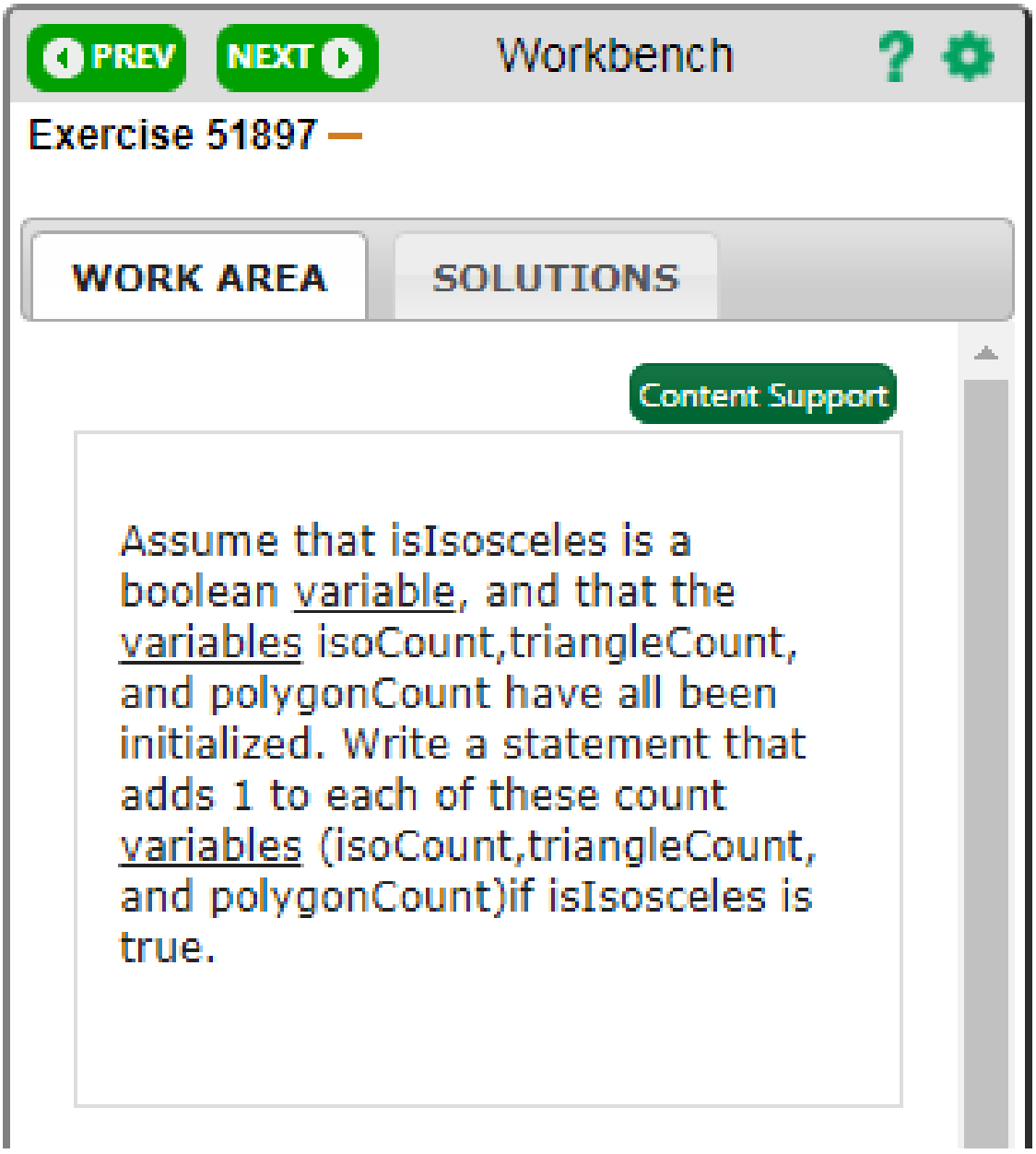
Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank

Programming Exercise



Programming Exercise Worksheet

this page is intentionally left blank