Calculating with numbers

Learning Objectives

- Learn to perform calculations with the help of variables.
- Know how to form printable statements including both calculations (expressions) and strings.

The basic mathematical operations are both familiar and straightforward: addition `+`, subtraction `-`, multiplication `*`, and division `/`. The precedence is also familiar: operations are performed from left to right with the parentheses taken into account. Expressions involving `*` and `/` are calculated before those involving `+` and `-`, as is customary in elementary school mathematics.

```java
int first = 2;
System.out.println(first); // prints 2
int second = 4;
System.out.println(second); // prints 4

int sum = first + second; // The sum of the values of the variables first and s
System.out.println(sum); // prints 6
```

Precedence and Parentheses

You can affect the order of operations by using parentheses. Operations within parentheses are performed before those outside them.

```java
int calculationWithParentheses = (1 + 1) + 3 * (2 + 5);
System.out.println(calculationWithParentheses); // prints 23
```

```
int calculationWithoutParentheses = 1 + 1 + 3 * 2 + 5;
System.out.println(calculationWithoutParentheses); // prints 13
```

The example above can also be divided into steps.

```
int calculationWithParentheses = (1 + 1);
System.out.println(calculationWithParentheses); // prints 2
calculationWithParentheses = calculationWithParentheses + 3 * (2 + 5);
System.out.println(calculationWithParentheses); // prints 23

int calculationWithoutParentheses = 1 + 1;
calculationWithoutParentheses = calculationWithoutParentheses + 3 * 2;
calculationWithoutParentheses = calculationWithoutParentheses + 5;
System.out.println(calculationWithoutParentheses); // prints 13
```

Programming exercise:

Seconds in a day

Points

1/1

In the exercise template, implement a program that asks the user for the number of days. After that, the program prints the number of seconds in the given number of days.

Earlier we learned to read an integer in the following manner:

```
Scanner scanner = new Scanner(System.in);

System.out.println("Give a number:");
int number = Integer.valueOf(scanner.nextLine());
System.out.println("You gave " + number);
```

Examples of expected output:

Sample output

```
How many days would you like to convert to seconds?
1
86400
```

Sample output

```
How many days would you like to convert to seconds?
3
259200
```

Sample output

```
How many days would you like to convert to seconds?
7
604800
```

Exercise submission instructions

How to see the solution

Quiz:

What does the program print?

Points:

1/1

What does the program below print?

```
int first = 10;
int second = first + 5;
first = 20;

System.out.println(second);
```

Select the correct answer

10

[✓ 15]

20

25

The answer is correct

Answered Tries remaining: 1

Expression and Statement

An *expression* is a combination of values that is turned into another value through a calculation or evaluation. The *statement* below includes the expression `1 + 1 + 3 * 2 + 5`, which is evaluated prior to its assignment to the variable.

```
int calculationWithoutParentheses = 1 + 1 + 3 * 2 + 5;
```

The evaluation of an expression is always performed before its value is assigned to a variable. As such, the calculation "1 + 1 + 3 * 2 + 5" in the example above is performed before the result is assigned to the variable.

An expression is evaluated where it occurs in the program's source code. As such, the evaluation can occur within a print statement, if the expression is used in calculating the value of the print statement's parameter.

```
int first = 2;
int second = 4;

System.out.println(first + second); // prints 6
System.out.println(2 + second - first - second); // prints 0
```

An expression does not change the value stored in a variable unless the expression's result is assigned to a variable or used as a parameter, for instance, in printing.

```java
int first = 2;
int second = 4;

// the expression below does not even work, since
// the result is not assigned to any variable
// or given as a parameter to a print statement
first + second;
```

Quiz:

Expressions and variable value

Points:

1/1

What does the following program print?

```java
int age = 32;
int height = 178;
height = age + 10;

System.out.println(age);
```

Select the correct answer

[✓ 32]

178

188

42

Correct! When the value of the variable `height` is set to the value of the `age` variable plus 10, the value of the variable `age` does not change.

Answered Tries remaining: 1

Calculating and Printing

The command `System.out.println` prints the value of a variable. The string literal to be printed, which is marked by quotation marks, can be appended with other content by using the operation `+`.

```
int length = 42;
System.out.println("Length " + length);
```

Sample output

```
Length 42
```

```
System.out.println("here is an integer --> " + 2);
System.out.println(2 + " <-- here is an integer");
```

Sample output

```
here is an integer —> 2
2 <— here is an integer
```

If one of the operands of the operation `+` is a string, the other operand will be changed into a string too during program execution. In the example below, the integer `2` is turned into the string "2", and a string has been appended to it.

The precedence introduced earlier also apply here:

```
System.out.println("Four: " + (2 + 2));
System.out.println("But! Twenty-two: " + 2 + 2);
```

Sample output

```
Four: 4
But! Twenty-two: 22
```

```java
public class Example {
    public static void main(String[] args) {
        System.out.println("Four: " + (2 + 2));
        System.out.println("But! Twenty-two: " + 2 + 2);
    }
}
```

main:3
Name Value

Output

Prev 1 / 4 Next

Programming exercise:

Sum of two numbers

Points 1/1

Write a program that asks the user for two numbers. After this, the program prints the sum of the numbers given by the user.

When you ask for multiple numbers, create a separate variable for each:

```java
Scanner scanner = new Scanner(System.in);

System.out.println("Give the first number:");
int first = Integer.valueOf(scanner.nextLine());
System.out.println("Give the second number:");
int second = Integer.valueOf(scanner.nextLine());
```

```
// do something with the numbers
```

Here's how the program is expected to work:

Sample output

```
Give the first number:
8
Give the second number:
3
The sum of the numbers is 11
```

Sample output

```
Give the first number:
3
Give the second number:
-1
The sum of the numbers is 2
```

Exercise submission instructions

How to see the solution

Programming exercise:

Sum of three numbers

Points 1/1

Write a program that asks the user for three numbers. After this the program prints the sum of the numbers given by the user.

The program should work like this:

Sample output

```
Give the first number:
8
```

```
Give the second number:
3
Give the third number:
3
The sum of the numbers is 14
```

Sample output

```
Give the first number:
3
Give the second number:
-1
Give the third number:
2
The sum of the numbers is 4
```

Exercise submission instructions

How to see the solution

Applying this knowledge, we can create an expression consisting of some text and a variable, which is evaluated in connection with the printing:

```java
int x = 10;

System.out.println("The value of the variable x is: " + x);

int y = 5;
int z = 6;

System.out.println("y is " + y + " and z is " + z);
```

Sample output

```
The value of the variable x is: 10
y is 5 and z is 6
```

Programming exercise:

Addition formula

Points

1/1

Create a program that can be used to add two integers together. In the beginning, the user is asked to give two integers that are to be summed. The program then prints the formula that describes the addition of the numbers.

Example output:

Sample output

```
Give the first number:
5
Give the second number:
4
5 + 4 = 9
```

Sample output

```
Give the first number:
73457
Give the second number:
12888
73457 + 12888 = 86345
```

Exercise submission instructions

How to see the solution

Programming exercise:

Multiplication formula

Points

1/1

Similar to the previous exercise, create a program that multiplies the values stored in two integer variables.

For instance, if the entered numbers are 2 and 8, the program should print the following:

Sample output

```
Give the first number:
2
Give the second number:
8
2 * 8 = 16
```

Likewise, if the entered numbers are 277 and 111, the print should be the following:

Sample output

```
Give the first number:
277
Give the second number:
111
277 * 111 = 30747
```

Exercise submission instructions

How to see the solution

Once you have completed the previous exercise, try finding out the greatest possible multiplication that you can calculate. The reason behind the phenomenon you'll observe is that the value of an integer value is capped at the maximum of 2^{31}-1 (i.e. 2147483647). This is because integer variables are represented with 32 bits in the computer's memory. Variable representation is covered in more detail on the Computer Organization course.

Division

Division of integers is a slightly trickier operation. The types of the variables that are part of the division determine the type of result achieved by executing the command. If all of the variables in the division expression are integers, the resulting value is an integer as well.

```
int result = 3 / 2;
System.out.println(result);
```

Sample output

```
1
```

The previous example prints 1: both 3 and 2 are integers, and the division of two integers always produces an integer.

```
int first = 3;
int second = 2;
double result = first / second;
System.out.println(result);
```

Sample output

```
1
```

The output 1 again, since first and second are (still) integers.

If the dividend or divisor (or both!) of the division is a floating point number, the result is a floating point number as well.

```
double whenDividendIsFloat = 3.0 / 2;
System.out.println(whenDividendIsFloat); // prints 1.5

double whenDivisorIsFloat = 3 / 2.0;
System.out.println(whenDivisorIsFloat); // prints 1.5
```

Sample output

```
1.5
1.5
```

An integer can be converted into a floating point number by placing a type-casting operation `(double)` before it:

```java
int first = 3;
int second = 2;

double result1 = (double) first / second;
System.out.println(result1); // prints 1.5

double result2 = first / (double) second;
System.out.println(result2); // prints 1.5

double result3 = (double) (first / second);
System.out.println(result3); // prints 1.0
```

Sample output

```
1.5
1.5
1.0
```

The last example produces an incorrectly rounded result, because the integer division is executed before the type casting.

If the result of a division is assigned to an integer-type variable, the result is automatically an integer.

```java
int integer = (int) 3.0 / 2;
System.out.println(integer);
```

Sample output

```
1
```

The next example prints "1.5"; the dividend is converted into a floating-point number by multiplying it with a floating-point number prior to executing the division.

```java
int dividend = 3;
int divisor = 2;
```

```
double result = 1.0 * dividend / divisor;
System.out.println(result);
```

Sample output

```
1.5
```

Calculating the average

The next exercises task you with calculating the average of the entered numbers. Let's briefly review the concept of *average*.

An average refers to the sum of numbers divided by their count. For instance, the average of the numbers 5 and 3 can be calculated with the formula (5+3)/2. Similarly, the average of the numbers 1, 2, and 4 is produced by the formula (1+2+4)/3.

In the context of programming, there are a few things to keep in mind. Firstly, dividing by zero is typically not permitted. This implies that calculating the average of zero numbers is impossible. Secondly, if the program handles both the sum of the numbers and their total count as integers, one (or both) of the variables should be casted to a floating-point number by multiplying it by 1.0 before the division.

Programming exercise:

Average of two numbers

Points

1/1

Write a program that asks the user for two integers and prints their average.

Sample output

```
Give the first number:
8
Give the second number:
```

```
2
The average is 5.0
```

Exercise submission instructions

How to see the solution

Programming exercise:

Average of three numbers

Points 1/1

Write a program that asks the user for three integers and prints their average.

Sample output

```
Give the first number:
8
Give the second number:
2
Give the third number:
3
The average is 4.333333333333333
```

Sample output

```
Give the first number:
9
Give the second number:
5
Give the third number:
-1
The average is 4.333333333333333
```

Exercise submission instructions

How to see the solution

Quiz:
Division

Points:
1/1

What does the program below print?

```
int dividend = 3;
int divisor = 2;

double result = dividend / divisor * 1.0;
System.out.println(result);
```

3.0

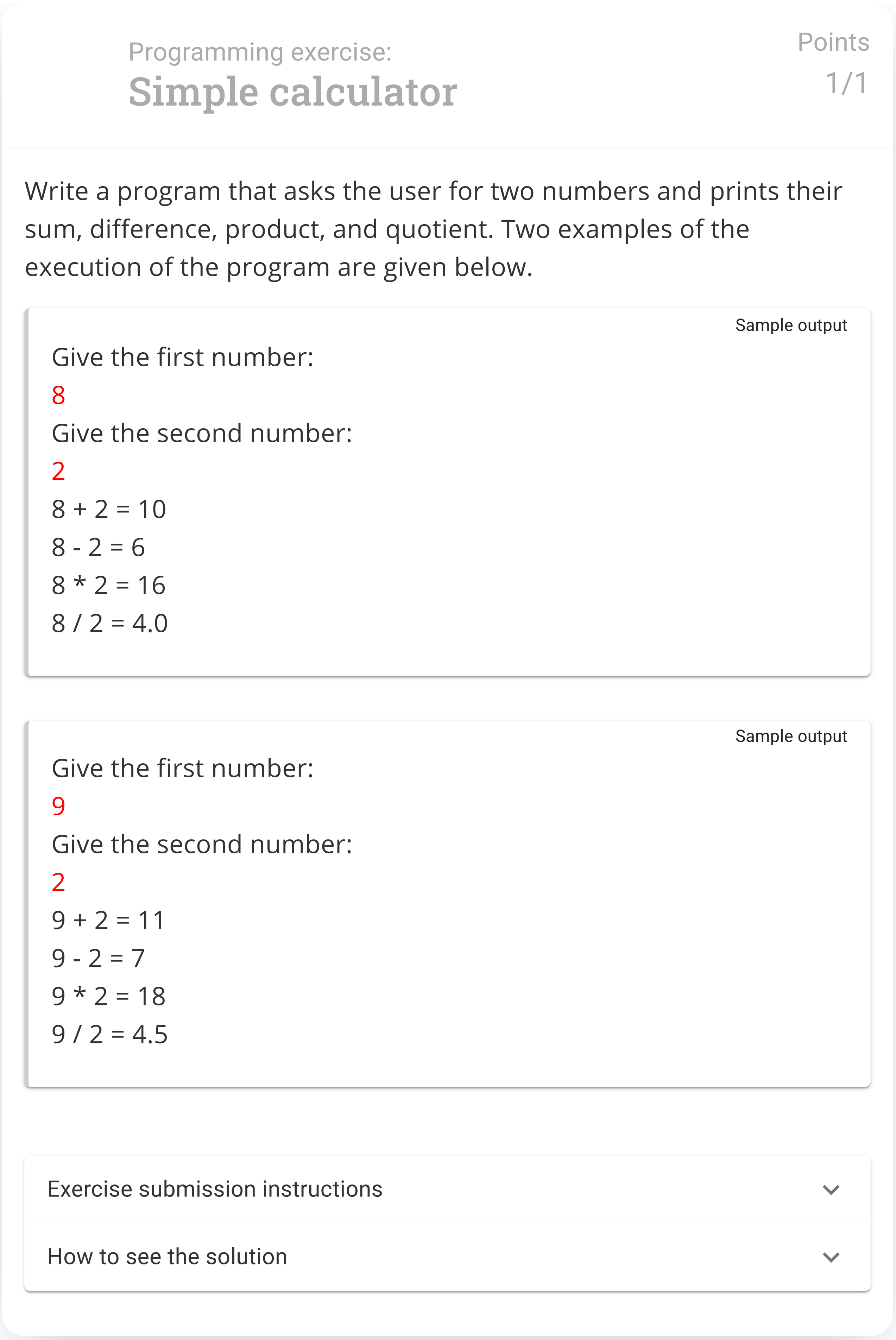
2.0

1.5

[✓ 1.0]

Correct!. The calculation dividend / divisor is executed first. Since both variables are integers, so is the result.

The answer is correct



Programming exercise:

Simple calculator

Points 1/1

Write a program that asks the user for two numbers and prints their sum, difference, product, and quotient. Two examples of the execution of the program are given below.

Sample output

```
Give the first number:
8
Give the second number:
2
8 + 2 = 10
8 - 2 = 6
8 * 2 = 16
8 / 2 = 4.0
```

Sample output

```
Give the first number:
9
Give the second number:
2
9 + 2 = 11
9 - 2 = 7
9 * 2 = 18
9 / 2 = 4.5
```

Misunderstandings Related to the Value of a Variable

When a computer executes program code, it does it one command at a time, always advancing exactly as specified by the code. When a value is assigned to a variable, the same chain of events always occurs: the value on the right-hand side of the equality sign is copied and assigned to the variable on the left-hand side (i.e., copied to the location specified by that variable).

It's crucial for a programmer to understand that assigning a value to a variable always does the same thing.

Here's three common misunderstandings related to assigning a value to a variable:

- Viewing value assignment as a transfer instead of a copy operation: once `first = second` has been executed, it's often assumed that the value of the variable `second` has been moved to the value of the variable `first`, and that the variable `second` no longer holds a value, or that its value is 0, for instance. This is incorrect, as executing `first = second` means that the value in the position specified by `second` is merely copied to the place specified by the variable `first`. Hence, the variable `second` is not modified.
- Viewing value assignment as creating a dependency instead of being a copy operation: once `first = second` has been executed, it's often assumed that any change in the value of the variable `second` is automatically also reflected in the value of the variable `first`. This is incorrect; assignment — i.e., copying — is a one-off event. It only occurs when the program code `first = second` is executed.
- The third misunderstanding concerns the direction of copying: it's often thought that in executing `first = second` the value of the variable `second` is set as the value of the variable `first`. The confusion also manifests itself in situations where the programmer accidentally writes e.g. `42 = value` — fortunately, IDEs provide support on this issue too.

Perhaps the best way to understand a program's execution flow is through the use of pen and paper. As you're reading the program, write down the names of any new variables, while making a note of how the values of the variables in the code change line by line. Let's demonstrate the execution with the following program code:

```
line 1: int first = (1 + 1);
line 2: int second = first + 3 * (2 + 5);
line 3:
line 4: first = 5;
line 5:
```

```java
line 6: int third = first + second;
line 7: System.out.println(first);
line 8: System.out.println(second);
line 9: System.out.println(third);
```

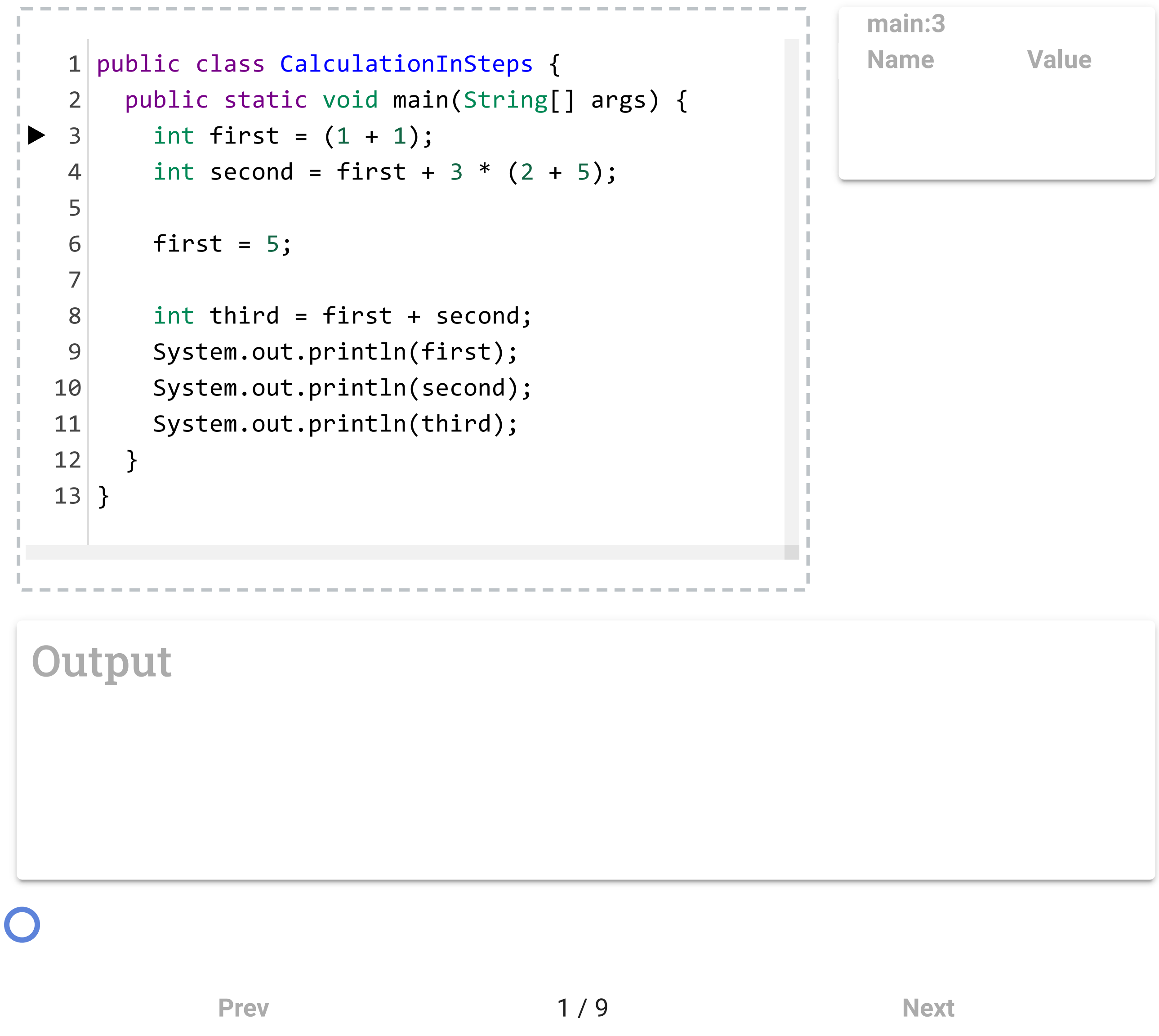
The execution flow of the program above has been broken down below.

Sample output

```
line 1: initiate a variable first
line 1: copy the result of the calculation 1 + 1 as the value of the variable first
line 1: the value of the variable first is 2
line 2: create the variable second
line 2: calculate 2 + 5, 2 + 5 -> 7
line 2: calculate 3 * 7, 3 * 7 -> 21
line 2: calculate first + 21
line 2: copy the value of the variable first into the calculation, its value is 2
line 2: calculate 2 + 21, 2 + 21 -> 23
line 2: copy 23 to the value of the variable second
line 2: the value of the variable second is 23
line 3: (empty, do nothing)
line 4: copy 5 to the value of the variable first
line 4: the value of the variable first is 5
line 5: (empty, do nothing)
line 6: create variable third
line 6: calculate first + second
line 6: copy the value of the variable first into the calculation, its value is 5
line 6: calculate 5 + second
line 6: copy the value of the variable second into the calculation, its value is 23
line 6: calculate 5 + 23 -> 28
line 6: copy 28 to the value of the variable third
line 6: the value of the variable third is 28
line 7: print the variable first
line 7: copy the value of the variable first for the print operation, its value is 5
line 7: print the value 5
```

line 8: print the variable second
line 8: copy the value of the variable second for the print operation, its value is 23
line 8: print the value 23
line 9: print the variable third
line 9: copy the value of the variable third for the print operation, its value is 28
line 9: we print the value 28

You'll find a step-by-step visualization of the previous program below, which goes through the program line by line — on each step, reflect on how the program ends up with the result that it prints.



```java
public class CalculationInSteps {
  public static void main(String[] args) {
    int first = (1 + 1);
    int second = first + 3 * (2 + 5);

    first = 5;

    int third = first + second;
    System.out.println(first);
    System.out.println(second);
    System.out.println(third);
  }
}
```

Quiz:

Copying variable value

Points:

1/1

What does the program below print?

```
int second = 32;
int first = 25;
second = 17;
first = second;

System.out.println(second);
```

Select the correct answer

32

[✓ 17]

25

Correct! The assignment "first = second;" does not affect the value of the variable `second`.

Answered Tries remaining: 1

You have reached the end of this section! Continue to the next section:

→ 6. Conditional statements and conditional operation

Remember to check your points from the ball on the bottom-right corner of the material!

In this part: