

Teaching Unit 1: Primitive Types and Expressions

1.1 Primitive Expressions

Solutions

1.1.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
int n = 4;  
n += 3 * 2;  
System.out.println(n - 1);
```

Output

9

Multiplication produces 6, compound assignment changes n to 10, and subtraction prints 9.

1.1.2 - Write code

Given `int earned = 17` and `int possible = 20`, calculate the percentage as a double and print it. Preserve fractional division.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
int earned = 17;  
int possible = 20;  
double percent = 100.0 * earned / possible;  
System.out.println(percent);
```

Teaching Unit 1: Primitive Types and Expressions

1.2 Integer Division and Mod

Solutions

1.2.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
System.out.println(-17 / 5);  
System.out.println(-17 % 5);
```

Output

```
-3  
-2
```

Truncating -3.4 gives -3, and $-17 = 5 * (-3) + (-2)$.

1.2.2 - Write code

Given the nonnegative three-digit integer $n = 582$, calculate and print the sum of its digits without converting it to a String.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
int n = 582;  
int sum = n / 100 + (n / 10) % 10 + n % 10;  
System.out.println(sum);
```

Teaching Unit 1: Primitive Types and Expressions

1.3 Casting and Math.random

Solutions

1.3.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
System.out.println((int) -3.9);  
System.out.println((double) (11 / 4));
```

Output

```
-3  
2.0
```

Narrowing truncates toward zero. The parentheses perform integer division before widening.

1.3.2 - Write code

Write an expression for a random integer from 4 through 9 inclusive. Show its smallest and largest possible results. Use $r = 0.0$ and $r = 0.999999$ to test the endpoints of the expression.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
double r = 0.0;  
System.out.println((int) (r * 6) + 4);  
r = 0.999999;  
System.out.println((int) (r * 6) + 4);  
// In use: (int) (Math.random() * 6) + 4
```

Teaching Unit 1: Primitive Types and Expressions

1.4 String Concatenation

Solutions

1.4.1 - Trace

Determine the exact output. Explain the Java rule or state changes that produce it.

```
System.out.println("A" + (3 + 4) + 2);
```

Output

A72

The parenthesized sum becomes 7; the remaining operations concatenate.

1.4.2 - Write code

Given `int correct = 8` and `int bonus = 3`, print exactly `Score: 11`. Make the numeric addition happen before concatenation.

One valid solution is shown below. Other implementations satisfying the contract are also valid.

```
int correct = 8;  
int bonus = 3;  
System.out.println("Score: " + (correct + bonus));
```