

Due to the print book page limit, we cannot include all good CheckPoint questions in the physical book. The CheckPoint on this Website may contain extra questions not printed in the book. The questions in some sections may have been reordered as a result. Nevertheless, it is easy to find the CheckPoint questions in the book on this Website. Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate the book, edition, and question number in your email.

Thanks!

Chapter 2 Check Point Questions

Section 2.2

▼ 2.2.1

Identify and fix the errors in the following code:

```
1 public class Test {
2     public void main(string[] args) {
3         double i = 50.0;
4         double k = i + 50.0;
5         double j = k + 1;
6
7         System.out.println("j is " + j + " and
8         k is " + k);
9     }
10 }
```

Line 2: Missing static for the main method.

Line 2: string should be String.

Lines 7-8: The string cannot be broken into two lines.

Hide Answer

Section 2.3

▼ 2.3.1

How do you write a statement to let the user enter a double value from the keyboard?
What happens if you entered 5a when executing the following code?

```
double radius = input.nextDouble();
```

Use

```
Scanner input = new Scanner(System.in);
double value = input.nextDouble();
```

A runtime error will occur if you entered 5a when executing the following code:

```
double radius = input.nextDouble();
```

Hide Answer

▼ 2.3.2

Are there any performance differences between the following two import statements?

```
import java.util.Scanner;
import java.util.*;
```

No

[Hide Answer](#)

Section 2.4

▼ 2.4.1

Which of the following identifiers are valid? Which are Java keywords?

miles, Test, a++, --a, 4#R, \$4, #44, apps
class, **public**, **int**, x, y, radius

Valid identifiers:

miles, Test, \$4, apps, x, y, radius

Invalid identifiers:

a++, --a, 4#R, #44, class, public, int

Keywords:

class, public, int

[Hide Answer](#)

Section 2.5

▼ 2.5.1

Identify and fix the errors in the following code:

```
1 public class Test {
2     public static void main(String[] args) {
3         int i = k + 2;
4         System.out.println(i);
5     }
6 }
```

Line 3: k is undefined.

[Hide Answer](#)

Section 2.6

▼ 2.6.1

Identify and fix the errors in the following code:

```
1 public class Test {
2     public static void main(String[] args) {
3         int i = j = k = 2;
4         System.out.println(i + " " + j + " " + k);
5     }
6 }
```

Note that the statement `int i = j = k = 2` in line 3 only declares i. j and k are not declared. The following line would declare i, j, and k:

`int i, j, k;`

To fix the error, change line 3 to

```
int j, k;  
int i = j = k = 2;
```

or

```
int i = 2;  
int j = 2;  
int k = 2;
```

or

```
int i = 2, j = 2, k = 2;
```

Hide Answer

Section 2.7

▼ 2.7.1

What are the benefits of using constants? Declare an int constant SIZE with value 20.

There are three benefits of using constants:

- (1) you don't have to repeatedly type the same value;
- (2) the value can be changed in a single location, if necessary;
- (3) the program is easy to read.

```
final int SIZE = 20;
```

Hide Answer

▼ 2.7.2

Translate the following algorithm into Java code:

Step 1: Declare a double variable named miles with initial value 100.

Step 2: Declare a double constant named KILOMETERS_PER_MILE with value 1.609.

Step 3: Declare a double variable named kilometers, multiply miles and KILOMETERS_PER_MILE, and assign the result to kilometers.

Step 4: Display kilometers to the console.

What is kilometers after Step 4?

```
double miles = 100;  
final double KILOMETERS_PER_MILE = 1.609;  
double kilometers = KILOMETERS_PER_MILE * miles;  
System.out.println(kilometers);
```

The value of kilometers is 160.9.

Hide Answer

Section 2.8

▼ 2.8.1

What are the naming conventions for class names, method names, constants, and variables? Which of the following items can be a constant, a method, a variable, or a class according to the Java naming conventions?

MAX_VALUE, Test, read, readDouble

Class names: Capitalize the first letter in each name.

Variables and method names: Lowercase the first word, capitalize the first letter in all subsequent words.

Constants: Capitalize all letters.

Constant: MAX_VALUE

Method or variable: read , readDouble

Class name: Test

Hide Answer

Section 2.9

▼ 2.9.1

Find the largest and smallest byte, short, int, long, float, and double. Which of these data types requires the least amount of memory?

For byte, from -128 to 127, inclusive.

For short, from -32768 to 32767, inclusive.

For int, from -2147483648 to 2147483647, inclusive.

For long, from -9223372036854775808 to 9223372036854775807.

For float, the smallest positive float is 1.40129846432481707e-45 and the largest float is 3.40282346638528860e+38.

For double, the smallest positive double is 4.94065645841246544e-324 and the largest double is 1.79769313486231570e+308d.

Hide Answer

▼ 2.9.2

Show the result of the following remainders.

56 % 6

78 % -4

-34 % 5

-34 % -5

5 % 1

1 % 5

2

2

-4

-4

0

1

Hide Answer

▼ 2.9.3

If today is Tuesday, what will be the day in 100 days?

$(2 + 100) \% 7 = 4$. So it is Thursday.

Hide Answer

▼ 2.9.4

What is the result of $25 / 4$? How would you rewrite the expression if you wished the result to be a floating-point number?

$25 / 4$ is 6. If you want the quotient to be a floating-point number, rewrite it as $25.0 / 4.0$, $25.0 / 4$, or $25 / 4.0$.

Hide Answer

▼ 2.9.5

Show the result of the following code:

```
System.out.println(2 * (5 / 2 + 5 / 2));
System.out.println(2 * 5 / 2 + 2 * 5 / 2);
System.out.println(2 * (5 / 2));
System.out.println(2 * 5 / 2);
```

8
10
4
5

Hide Answer

▼ 2.9.6

Are the following statements correct? If so, show the output.

```
System.out.println("25 / 4 is " + 25 / 4);
System.out.println("25 / 4.0 is " + 25 / 4.0);
System.out.println("3 * 2 / 4 is " + 3 * 2 / 4);
System.out.println("3.0 * 2 / 4 is " + 3.0 * 2 / 4);
```

Yes, the statements are correct. The output is

25 / 4 is 6
25 / 4.0 is 6.25
3 * 2 / 4 is 1
3.0 * 2 / 4 is 1.5

Hide Answer

▼ 2.9.7

Write a statement to display the result of $2^{3.5}$.

`Math.pow(2, 3.5)`

Hide Answer

▼ 2.9.8

Suppose m and r are integers. Write a Java expression for mr^2 to obtain a floating-point result.

`1.0 * m * (r * r)`

Hide Answer

Section 2.10

▼ 2.10.1

How many accurate digits are stored in a float or double type variable?

A float value has 7-8 number of accurate digits and a double value has 15-17 number of accurate digits.

Hide Answer

▼ 2.10.2

Which of the following are correct literals for floating-point numbers?

12.3, 12.3e+2, 23.4e-2, -334.4, 20.5, 39F, 40D

All can be used as literals for floating-point numbers.

Hide Answer

▼ 2.10.3

Which of the following are the same as 52.534?

5.2534e+1, 0.52534e+2, 525.34e-1, 5.2534e+0

5.2534e+1, 0.52534e+2, 525.34e-1 are the same as 52.534.

Hide Answer

▼ 2.10.4

Which of the following are correct literals?

5_2534e+1, _2534, 5_2, 5_

5_2534e+1, and 5_2, are correct.

Hide Answer

Section 2.11

▼ 2.11.1

How would you write the following arithmetic expression in Java?

a.
$$\frac{4}{3(r + 34)} - 9(a + bc) + \frac{3 + d(2 + a)}{a + bd}$$

b. $5.5 * (r + 2.5)^{2.5 + t}$

(a) $4.0 / (3 * (r + 34)) - 9.0 / (a + b * c) + (3 + d * (2 + a)) / (a + b * d)$

(b) $5.5 * \text{Math.pow}(r + 2.5, 2.5 + t)$

Hide Answer

Section 2.12

▼ 2.12.1

How do you obtain the current second, minute, and hour?

`long totalMills = System.currentTimeMillis()` returns the milliseconds since Jan 1, 1970.
`long totalSeconds = totalMills / 1000` returns the total seconds.
`long totalMinutes = totalSeconds / 60` returns the total minutes.
`totalSeconds % 60` returns the current second.
`totalMinutes % 60` returns the current minute.
`totalMinutes / 60 % 24` returns the current hour.

Hide Answer

Section 2.13

▼ 2.13.1

Show the output of the following code:

```
double a = 6.5;
a += a + 1;
System.out.println(a);
a = 6;
a /= 2;
System.out.println(a);
```

14.0
3.0

Hint: `a += a + 1` is `a += 6.5 + 1`, `a += 7.5`, `a = a + 7.5`, `a = 6.5 + 7.5`. Therefore, `a` is 14.0.

Hide Answer

Section 2.14

▼ 2.14.1

Which of these statements are true?

- a. Any expression can be used as a statement.
- b. The expression `x++` can be used as a statement.
- c. The statement `x = x + 5` is also an expression.
- d. The statement `x = y = x = 0` is illegal.

b and c are true.

Hide Answer

▼ 2.14.2

Show the output of the following code:

```
int a = 6;
int b = a++;
System.out.println(a);
System.out.println(b);
a = 6;
b = ++a;
System.out.println(a);
System.out.println(b);
```

7
6
7
7

Hide Answer

Section 2.15

▼ 2.15.1

Can different types of numeric values be used together in a computation?

Yes. Different types of numeric values can be used in the same computation through numeric conversions referred to as casting.

Hide Answer

▼ 2.15.2

What does an explicit casting from a double to an int do with the fractional part of the double value? Does casting change the variable being cast?

The fractional part is truncated. Casting does not change the variable being cast.

Hide Answer

▼ 2.15.3

Show the output of the following code:

```
float f = 12.5F;
int i = (int)f;
System.out.println("f is " + f);
System.out.println("i is " + i);
```

```
f is 12.5
i is 12
```

Hide Answer

▼ 2.15.4

If you change `(int)(tax * 100) / 100.0` to `(int)(tax * 100) / 100` in line 11 in Listing 2.8, what will be the output for the input purchase amount of 197.556?

The answer is 11

Here is the reason:

```
tax = purchaseAmount * 0.06 = 197.556 * 0.06 = 11.85336
tax * 100 = 1185.336
(int)(tax * 100) = 1185
1185 / 100 = 11
```

Hide Answer

▼ 2.15.5

Show the output of the following code:

```
double amount = 5;
System.out.println(amount / 2);
System.out.println(5 / 2);
```

```
2.5
2
```


Hide Answer

▼ 2.15.6

Write an expression that rounds up a double value in variable `d` to an integer.

`(int)(d + 0.5)`

Hide Answer

Section 2.16

▼ 2.16.1

How would you write the following arithmetic expression?

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

`(b + Math.pow(b * b - 4 * a * c, 0.5)) / (2 * a)`

Hide Answer

Section 2.17

▼ 2.17.1

Show the output in Listing 2.10 with the input value 1.99.

```
Enter an amount in double, for example 11.56: 1.99
Your amount 1.99 consists of
    1 dollars
    3 quarters
    2 dimes
    0 nickels
    4 pennies
```

Hide Answer

Section 2.18

▼ 2.18.1

Can you declare a variable as `int` and later redeclare it as `double`?

No.

Hide Answer

▼ 2.18.2

What is an integer overflow? Can floating-point operations cause overflow?

Numbers are stored with a limited numbers of digits. When a variable is assigned a value that is too large (in size) to be stored, it causes overflow. Overflow is for integer operations. Floating-point operations will not cause overflow.

Hide Answer

▼ 2.18.3

Will overflow cause a runtime error?

No.

Hide Answer

▼ **2.18.4**

What is a round-off error? Can integer operations cause round-off errors? Can floating-point operations cause round-off errors?

A round-off error, also called a rounding error, is the difference between the calculated approximation of a number and its exact mathematical value. Integer operations will not cause rounding error. Floating-point operations may cause rounding error.

Hide Answer