



Java Magazine

Java SE, Quiz

Quiz yourself: Declaring and initializing one-dimensional arrays in Java



[Simon Roberts](#) | January 24, 2022



[Mikalai Zaikin](#) | January 24, 2022



There are several ways to create an array and initialize an array declaration. Some can be used anywhere. Others can't.

Which code will print 0 to console? Choose one.

- A.

```
int arr1[] = new int[10];  
System.out.print(arr1[0]);
```
- B.

```
int arr2[10] = new int[10];  
System.out.print(arr2[3]);
```
- C.

```
int[] arr3 = new int[0]{};  
System.out.print(arr3.length);
```

The answer is A.

The answer is B.

The answer is C.

D. `int[] arr4 = {0,0};`
`System.out.print(arr4[arr4.length]);`

The answer is D.

Answer. This question explores the declaration and initialization of one-dimensional arrays and the syntax for declaring variables of array type.

To declare a variable of single-dimension array type, two forms are possible, with the first being the more common. Of course, either of these forms can be modified with initialization before the final semicolon.

- `ElementType [] varName;`
- `ElementType varName [];`

In Java, arrays are objects, and they know their own *element count*. A variable that refers to an array object knows the type of the array but *not* the size, while the array object itself knows both its element type and count. Because of this, it is never permitted to include an element count in the variable declaration part, so the form used in option B is illegal and doesn't compile. Therefore, option B is incorrect.

To create an array (and potentially, therefore, to initialize an array declaration), three forms are possible.

- `new ElementType[elementCount]`
- `new ElementType[] { elementValue0, elementValue1, ... }`
- `{ elementValue0, elementValue1, ... }`

In the second and third forms, the ellipsis (...) indicates that any number of values may be specified in a comma-separated list. This potentially includes no elements at all.

The first and second forms can be used anywhere, while the third form can be used only to initialize a new declaration. That is, the following forms are permitted:

```
int [] ia = {0, 1, 2};  
// initializing a declaration
```

and

```
public static void doStuff(int [] ia) {}  
//----- elsewhere  
doStuff(new int [] {0, 1, 2});
```

But the following forms are *not* permitted:

```
int [] ia;  
  
ia = {0, 1, 2};  
// illegal – not initializing a declaration
```

and

```
public static void doStuff(int [] ia) {}  
//----- elsewhere  
doStuff({0, 1, 2});
```

```
// illegal – not initializing a declaration
```

Notice that only the first form permits a numeric dimension specification. In both the second and third forms, the number of elements is implicitly determined from the number of elements listed.

From these discussions you can see that option C, which attempts to use the curly brace format to enumerate the array elements with an explicit array size, is illegal and does not compile. Therefore, option C is incorrect.

At this point, consider options A and D. Both of these are syntactically correct and will compile. Let's see what happens next.

Option A demonstrates declaration and default initialization of an array of `int`. No explicit initialization occurs here, so the question is what values does the array contain?

Java performs a consistent default initialization of all objects, and arrays are objects even if they're special in many ways. Default initialization might be followed by other explicit initializations if the source calls for them, for example, running a constructor. In this question, however, the array is subject only to the default initialization.

So, what's the default initialization? Simple enough, right after the allocation of the memory for the object, the JVM ensures that the space contains *zero-like* values. Specifically, booleans are set to false, other primitives are set to zero, and any reference type is set to null. Since this array has an `int` base type, the 10 elements (at indexes 0 to 9) will all contain the value 0.

From this it's clear that the element that's printed in option A, that is, the element at index 0, contains the value 0, which makes option A correct.

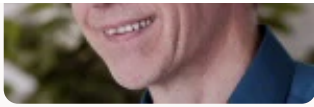
As mentioned earlier, the code in option D correctly declares and initializes an array. In this case, the two elements are explicitly initialized to the value 0. However, when the `print` statement is executed, it tries to access the array element at subscript 2 (because the length of the array is 2). This fails because the valid subscripts for a two-element array are 0 and 1. Consequently, the code in option D throws a `java.lang.ArrayIndexOutOfBoundsException` exception and, thus, option D is incorrect.

Conclusion. The correct answer is option A.

Related quizzes

- [Quiz yourself: Declare and use List and ArrayList instances](#)
- [Quiz yourself: One-dimensional arrays](#)
- [Quiz yourself: Two-dimensional arrays](#)





Simon Roberts

Simon Roberts joined Sun Microsystems in time to teach Sun's first Java classes in the UK. He created the Sun Certified Java Programmer and Sun Certified Java Developer exams. He wrote several Java certification guides and is currently a freelance educator who publishes recorded and live video training through Pearson InformIT (available direct and through the O'Reilly Safari Books Online service). He remains involved with Oracle's Java certification projects.



Mikalai Zaikin

Mikalai Zaikin is a lead Java developer at IBA IT Park in Minsk, Belarus. During his career, he has helped Oracle with development of Java certification exams, and he has been a technical reviewer of several Java certification books, including three editions of the famous *Sun Certified Programmer for Java* study guides by Kathy Sierra and Bert Bates.

[◀ Previous Post](#)

Resources for

About
Careers
Developers
Investors
Partners
Startups

Why Oracle

Analyst
Reports
Best CRM
Cloud
Economics
Corporate
Responsibility
Diversity and
Inclusion
Security
Practices

Learn

What is
Customer
Service?
What is ERP?
What is
Marketing
Automation?
What is
Procurement?
What is Talent
Management?
What is VM?

What's New

Try Oracle
Cloud Free Tier
Oracle
Sustainability
Oracle COVID-
19 Response
Oracle and
SailGP
Oracle and
Premier
League
Oracle and Red
Bull Racing
Honda

Contact Us

US Sales
1.800.633.0738
How can we help?
Subscribe to
Oracle Content
Try Oracle Cloud
Free Tier
Events
News

