



Practice ▾ Tutorials ▾ 30 Days of Code ▾ Day 21: Generics

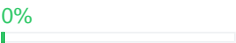
COMPETE

JOBS

LEADERBOARD

8 more challenges to get your gold badge!

Days of
Code



- ☐
- ☐
- ☐
- ☐ joel_h_healy ▾

You have successfully solved Day 21: Generics



Share

Tweet

You are now 8 challenges away from the gold level for your 30 days of code badge.

[Try the Next Challenge](#) | [Try a Random Challenge](#)

Day 21: Generics ▾

by Shafaet

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Objective

Today we're discussing Generics; be aware that *not all languages support this construct*, so fewer languages are enabled for this challenge. Check out the [Tutorial](#) tab for learning materials and an instructional video!

Task

Write a single generic function named *printArray*; this function must take an array of generic



Submitted 39705 times
Max Score 30

Need Help?

elements as a parameter (the exception to this is C++, which takes a *vector*). The locked *Solution* class in your editor tests your function.

Note: You must use generics to solve this challenge. *Do not* write overloaded functions.

Input Format

The locked *Solution* class in your editor will pass different types of arrays to your *printArray* function.

Constraints

- You must have exactly function named *printArray*.


Output Format

Your *printArray* function should print each element of its generic array parameter on a new line.

 [View Tutorial](#)

 [View Discussions](#)

 [View Editorial Solution](#)

 [View Top Submissions](#)

Rate This Challenge:

☐ ☐ ☐ ☐ ☐

[Download problem statement](#)

[Download sample test cases](#)

[Suggest Edits](#)



Current Buffer (saved locally, editable) ☐ ☐

Java 7 ☐ ☐

```
1 import java.io.*;
2
3 class Printer <T> {
4     /**
5      * Method Name: printArray
6      * Print each element of the generic array on a new line. Do not return anything.
7      * @param A generic array
8      */
9
10    // Write your code here
11    public static <T> void printArray(T[] array) {
12        for (T element : array) {
13            System.out.println(element);
14        }
15    }
```

```

16 }
17
18 public class Generics {
19
20     public static void main(String args[]){
21         Scanner scanner = new Scanner(System.in);
22         int n = scanner.nextInt();
23         Integer[] intArray = new Integer[n];
24         for (int i = 0; i < n; i++) {
25             intArray[i] = scanner.nextInt();
26         }
27
28         n = scanner.nextInt();
29         String[] stringArray = new String[n];
30         for (int i = 0; i < n; i++) {
31             stringArray[i] = scanner.next();
32         }
33
34         Printer<Integer> intPrinter = new Printer<Integer>();
35         Printer<String> stringPrinter = new Printer<String>();
36         intPrinter.printArray( intArray );
37         stringPrinter.printArray( stringArray );
38         if(Printer.class.getDeclaredMethods().length > 1){
39             System.out.println("The Printer class should only have 1 method named printArray.");
40         }
41     }
42 }

```

Line: 14 Col: 1

☐ [Upload Code as File](#)
☐ Test against custom input