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

COMPETE

JOBS

LEADERBOARD

8 more challenges to get your gold badge!

Days of Code

0%

joel_h_healy

✕

You have successfully solved Day 21: Generics 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!

Submitted 39705 times
Max Score 30

Need Help?

Task

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

```
import ↔;

class Printer <T> {
    /**
    *    Method Name: printArray
    *    Print each element of the generic array on a new line. Do not return anything.
    *    @param A generic array
    **/

    // Write your code here
    public static <T> void printArray(T[] array) {
        for (T element : array) {
            System.out.println(element);
        }
    }
```

```java
}

public class Generics {

    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        Integer[] intArray = new Integer[n];
        for (int i = 0; i < n; i++) {
            intArray[i] = scanner.nextInt();
        }

        n = scanner.nextInt();
        String[] stringArray = new String[n];
        for (int i = 0; i < n; i++) {
            stringArray[i] = scanner.next();
        }

        Printer<Integer> intPrinter = new Printer<Integer>();
        Printer<String> stringPrinter = new Printer<String>();
        intPrinter.printArray( intArray  );
        stringPrinter.printArray( stringArray );
        if(Printer.class.getDeclaredMethods().length > 1){
            System.out.println("The Printer class should only have 1 method named printArray.");
        }
    }
}
```

Line: 14 Col: 1

□ Test against custom input

Run Code

Submit Code