

See the Assessment Guide for information on how to interpret this report.

ASSESSMENT SUMMARY

Compilation: PASSED

API: PASSED

SpotBugs: PASSED

PMD: PASSED

Checkstyle: PASSED

Correctness: 13/13 tests passed

Memory: No tests available for autograding.

Timing: No tests available for autograding.

Aggregate score: 100.00%

[Compilation: 5%, API: 5%, Style: 0%, Correctness: 90%]

ASSESSMENT DETAILS

The following files were submitted:

298 Jun 13 21:55 HelloGoodbye.java

180 Jun 13 21:55 HelloWorld.java

500 Jun 13 21:55 RandomWord.java

```
*****
*   COMPILING
*****
```

```
% javac HelloWorld.java
```

```
*-----
```

```
% javac HelloGoodbye.java
```

```
*-----
```

```
% javac RandomWord.java
```

```
*-----
```

```
=====
```

Checking the APIs of your programs.

```
*-----
```

HelloWorld:

HelloGoodbye:

RandomWord:

```
=====
```

```
*****
```

```
* CHECKING STYLE AND COMMON BUG PATTERNS
```

```
*****
```

```
% spotbugs *.class
```

```
*-----
```

```
=====
```

```
% pmd .
```

```
*-----
```

```
=====
```

```
% checkstyle *.java
```

```
*-----
```

```
% custom checkstyle checks for HelloWorld.java
```

```
*-----
```

```
% custom checkstyle checks for HelloGoodbye.java
```

```
*-----
```

```
% custom checkstyle checks for RandomWord.java
```

```
*-----
```

```
=====
```

```
*****
```

```
* TESTING CORRECTNESS
```

```
*****
```

```
Testing correctness of HelloWorld
```

```
*-----
```

```
Running 2 total tests.
```

```
Test 1: check output format
```

```
% java HelloWorld
```

```
Hello, World
```



```
Test 2: check correctness
```

```
* java HelloWorld
```



```
HelloWorld Total: 2/2 tests passed!
```

```
=====
```

```
Testing correctness of HelloGoodbye
```

```
*-----
```

```
Running 6 total tests.
```

```
Test 1: check output format
```

```
% java HelloGoodbye Kevin Bob
Hello Kevin and Bob.
Goodbye Bob and Kevin.
```

```
% java HelloGoodbye Alejandra Bahati
Hello Alejandra and Bahati.
Goodbye Bahati and Alejandra.
```

==> passed

Test 2: check correctness using names from assignment specification

```
* java HelloGoodbye Kevin Bob
* java HelloGoodbye Alejandra Bahati
```

==> passed

Test 3: check correctness using fixed names

```
* java HelloGoodbye Chandra Deshi
* java HelloGoodbye Ayşe María
* java HelloGoodbye Wayan Taiyeo
* java HelloGoodbye Ástfríður Bedřiška
```

==> passed

Test 4: check correctness when two names are the same

```
* java HelloGoodbye Turing Turing
* java HelloGoodbye Lovelace Lovelace
* java HelloGoodbye Hopper Hopper
* java HelloGoodbye Knuth Knuth
```

==> passed

Test 5: check correctness using random names

```
* java HelloGoodbye Dietger Bergný
* java HelloGoodbye Ffion Erjeta
* java HelloGoodbye Daulet Leny
* java HelloGoodbye Yawar Herm
* java HelloGoodbye Ertunç Nekane
```

==> passed

Test 6: test correctness using many random names

```
* 10 pairs of random names
* 100 pairs of random names
* 1000 pairs of random names
* 10000 pairs of random names
```

==> passed

HelloGoodbye Total: 6/6 tests passed!

=====

Testing correctness of RandomWord

*-----

Running 5 total tests.

Test 1: check that programs prints exactly one word

```
% java-algs4 RandomWord < animals8.txt
bear
```

```
% java-algs4 RandomWord < animals8.txt
cat
```

```
% java-algs4 RandomWord < coin.txt
tails
```

```
% java-algs4 RandomWord < names.txt
Chelliah
```

==> passed

Test 2: check that the word the program prints appears on standard input

- * 10000 trials from coin.txt
- * 10000 trials from animals8.txt
- * 10000 trials from animals.txt
- * 10000 trials from colors.txt
- * 10000 trials from algorithms.txt
- * 10000 trials from dream.txt

==> passed

Test 3: check that program consumes all data from standard input

- * coin.txt
- * animals8.txt
- * animals.txt
- * colors.txt
- * algorithms.txt
- * names.txt
- * dream.txt

==> passed

Test 4: check that words printed are uniformly random

- * 12000 trials from coin.txt
- * 12000 trials from roshambo.txt
- * 12000 trials from die.txt
- * 12000 trials from animals8.txt
- * 12000 trials from animals25.txt
- * 12000 trials from algorithms.txt

==> passed

Test 5: check that words printed are uniformly random
(with some words on standard input repeated)

- * 36000 trials from biased-coin.txt
- * 36000 trials from loaded-die.txt
- * 36000 trials from two-dice.txt
- * 36000 trials from tinyTale.txt

==> passed

RandomWord Total: 5/5 tests passed!

=====