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

ASSESSMENT SUMMARY

```
Compilation:  PASSED
API:          PASSED

SpotBugs:     PASSED
PMD:          PASSED
Checkstyle:   FAILED (0 errors, 2 warnings)

Correctness:  25/25 tests passed
Memory:       3/3 tests passed
Timing:       No tests available for autograding.

Aggregate score: 100.00%
[ Compilation: 5%, API: 5%, Style: 0%, Correctness: 80%, Timing: 10% ]
```

ASSESSMENT DETAILS

```
The following files were submitted:
----------------------------------
1.7K Jul 18 20:17 Bar.java
1.8K Jul 18 20:17 BarChartRacer.java


********************************************************************************
*  COMPILING
********************************************************************************


% javac Bar.java
*-----------------------------------------------------------

% javac BarChartRacer.java
*-----------------------------------------------------------


================================================================


Checking the APIs of your programs.
*-----------------------------------------------------------
Bar:

BarChartRacer:

================================================================


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


% spotbugs *.class
*-----------------------------------------------------------


================================================================


% pmd .
*-----------------------------------------------------------


================================================================


% checkstyle *.java
*-----------------------------------------------------------

% custom checkstyle checks for Bar.java
*-----------------------------------------------------------

% custom checkstyle checks for BarChartRacer.java
*-----------------------------------------------------------
[WARN] BarChartRacer.java:1: Use the 'readLine()' and 'hasNextLine()' methods from the 'In' class to read the data file one line at a time. [
[WARN] BarChartRacer.java:1: Call 'Arrays.sort()' to sort the Bar objects by value. [Sort]
Checkstyle ends with 0 errors and 2 warnings.


================================================================


********************************************************************************
*  TESTING CORRECTNESS
```

```
********************************************************************************

Testing correctness of Bar
*-----------------------------------------------------------
Running 6 total tests.

Test 1: check getName()
  * 1000 random records from cities.txt
  * 1000 random records from brands.txt
  * 1000 random records from movies.txt
  * 1000 random records from football.txt
==> passed

Test 2: check getCategory()
  * 1000 random records from cities.txt
  * 1000 random records from brands.txt
  * 1000 random records from movies.txt
  * 1000 random records from football.txt
==> passed

Test 3: check getValue()
  * 1000 random records from cities.txt
  * 1000 random records from brands.txt
  * 1000 random records from movies.txt
  * 1000 random records from football.txt
==> passed

Test 4: check that compareTo() satisfies Comparable contract
  * reflexive     (0 <= value < 1000000)
  * reflexive     (0 <= value < 10)
  * antisymmetric (0 <= value < 1000000)
  * antisymmetric (0 <= value < 10)
  * transitive    (0 <= value < 1000000)
  * transitive    (0 <= value < 10)
  * consistent    (0 <= value < 1000000)
  * argument is null
==> passed

Test 5: check correctness of compareTo() for random values
  * 0 <= value < 1000000
  * 0 <= value < 10000
  * 0 <= value < 1000
  * 0 <= value < 100
  * 0 <= value < 10
==> passed

Test 6: call Bar constructor with invalid arguments
  * new Bar("Cairo", 1500, null)
  * new Bar("Cairo", -1, "Middle East")
  * new Bar(null, 1500, "Middle East")
  * new Bar("Vijayanagar", -2147483648, "South Asia")
  * new Bar(null, 500, null)
  * new Bar(null, -500, null)
==> passed


Total: 6/6 tests passed!


================================================================
********************************************************************************
*  MEMORY
********************************************************************************

Analyzing memory of Bar
*-----------------------------------------------------------
Running 3 total tests.

Test 1: memory of new Bar("Tokyo", 38194, "East Asia")
        [ must be <= 1.1x reference solution ]
  - memory of student   Bar = 176 bytes
  - memory of reference Bar = 176 bytes
  - student / reference     = 1.00
==> passed

Test 2: memory of new Bar("Mexico City", 21520, "Latin America")
        [ must be <= 1.1x reference solution ]
  - memory of student   Bar = 184 bytes
  - memory of reference Bar = 184 bytes
  - student / reference     = 1.00
==> passed

Test 3: memory of new Bar("Star Wars: The Force Awakens", 936662225, "Buena Vista")
        [ must be <= 1.1x reference solution ]
  - memory of student   Bar = 200 bytes
  - memory of reference Bar = 200 bytes
  - student / reference     = 1.00
==> passed


Total: 3/3 tests passed!

================================================================



********************************************************************************
```

```
*  TESTING CORRECTNESS (substituting reference Bar.java)
********************************************************************************

Testing correctness of BarChartRacer
*-----------------------------------------------------------
Running 19 total tests.

Test 1: check standard output format
  % java BarChartRacer cities.txt 10
  [no output]

  % java BarChartRacer brands-yearly.txt 12
  [no output]

  % java BarChartRacer movies.txt 15
  [no output]

==> passed

Test 2: count calls to methods in In
 * file = cities.txt           k = 10
 * file = brands.txt           k = 12
 * file = movies.txt           k = 15
==> passed

Test 3a: count calls to methods in StdDraw (only one group)
 * file = cities2018.txt       k = 10
 * file = cities1500.txt       k = 10
 * file = movies1982.txt       k = 15
 * file = brands2018.txt       k = 12
 * file = football2019.txt     k = 5
==> passed

Test 3b: count calls to methods in StdDraw (k = n)
 * file = cities.txt           k = 12
 * file = brands-yearly.txt    k = 100
==> passed

Test 3c: count calls to methods in StdDraw
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
==> passed

Test 4a: count calls to methods in BarChart (only one group)
 * file = cities2018.txt       k = 10
 * file = cities1500.txt       k = 10
 * file = movies1982.txt       k = 15
 * file = brands2018.txt       k = 12
 * file = football2019.txt     k = 5
==> passed

Test 4b: count calls to methods in BarChart (k = n)
 * file = cities.txt           k = 12
 * file = brands-yearly.txt    k = 100
==> passed

Test 4c: count calls to methods in BarChart
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
==> passed

Test 5: check title
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
==> passed

Test 6: check data source
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
==> passed

Test 7: check x-axis label
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
==> passed

Test 8a: check that correct bars are drawn (only one group)
 * file = cities2018.txt       k = 10
 * file = cities1500.txt       k = 10
 * file = movies1982.txt       k = 15
 * file = brands2018.txt       k = 12
 * file = football2019.txt     k = 5
==> passed

Test 8b: check that correct bars are drawn (k = n)
 * file = cities.txt           k = 12
 * file = brands-yearly.txt    k = 100
==> passed

Test 8c: check that correct bars are drawn (fixed k)
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
```

```
==> passed

Test 8d: check that correct bars are drawn (varying k)
 * file = cities.txt           1 <= k <= 12
 * file = patents-yearly.txt   1 <= k <= 50
 * file = brands-yearly.txt    1 <= k <= 100
==> passed

Test 9a: check that bars are drawn in descending order (only one group)
 * file = cities2018.txt       k = 10
 * file = cities1500.txt       k = 10
 * file = movies1982.txt       k = 15
 * file = brands2018.txt       k = 12
 * file = football2019.txt     k = 5
==> passed

Test 9b: check that bars are drawn in descending order (k = n)
 * file = cities.txt           k = 12
 * file = brands-yearly.txt    k = 100
==> passed

Test 9c: check that bars are drawn in descending order (fixed k)
 * file = cities.txt           k = 10
 * file = brands-yearly.txt    k = 12
 * file = movies.txt           k = 15
==> passed

Test 9d: check that bars are drawn in descending order (varying k)
 * file = cities.txt           1 <= k <= 12
 * file = patents-yearly.txt   1 <= k <= 50
 * file = brands-yearly.txt    1 <= k <= 100
==> passed


BarChartRacer Total: 19/19 tests passed!


================================================================
```