

Homework 2

Before attempting this project, be sure you have completed all of the reading assignments, hands-on labs, discussions, and assignments to date.

Design a Java class named Guitar that contains:

- A private int data field named numStrings that defines the number of strings on the guitar. The default value should be 6.
- A private double data field named guitarLength that defines the length of the guitar in inches. The default value should be 28.2
- A private String data field named guitarManufacturer that defines the manufacturer of the guitar. The default value should be "Gibson".
- A private Color data field named guitarColor that defines the color of the guitar. The default value should be Color.Red.
- A no argument constructor that creates a Guitar using the default number of strings, length, manufacturer and color.
- A constructor that creates a Guitar using a specified number of strings, length, manufacturer and color.
- Getter methods for all data fields.
- A playGuitar() method that returns a string representation of 16 randomly selected musical notes of random duration. For example, the first part of the string returned might look like this: [A(2), G(3), B(0.5), C(1), C(1), D(0.25), ...]. You can assume one octave in the key of C where valid notes include A, B, C, D, E, F and G and duration values are .25, .5, 1, 2, and 4 representing sixteenth notes, eighth notes, quarter notes, half notes and whole notes, respectively.
- A toString() method that displays the number of strings, length, manufacturer and color in String format

Be sure your code compiles.

Write a Java test program, named TestGuitar, to create 3 different Guitars representing each representing a unique test case and call each all of the getter methods along with the toString and playGuitar() methods and document the output. For example for a Guitar with 7 strings, length of 30.2, manufactured by Fender with a color of Black, the output may look similar to this:

*****Output*****

```
toString(): (numStrings=7, Length=30.2, manufacturer=Fender, color=Black)
getNumStrings(): 7
getGuitarLength(): 30.2
getGuitarManufacturer(): Fender
getGuitarColor(): Black
playGuitar(): [A(2), G(3), B(0.5), C(1), C(1), D(0.25), E(2), F(2), G(0.25), C(4), C(1), F(0.25),A(1), C(2),
D(4),C(4)]
```

Document your test cases in the form of table with columns indicating the input values, expected output, actual output and if the test case passed or failed. This table should contain 4 columns with

appropriate labels and a row for each test case. An example template is shown below. Note that the actual output should be the actual results you receive when running your program and applying the input for the test record.

Keep in mind, for three guitars, you will have three different output results. Also, note there is no requirement to actually play the notes of the guitar. The notes are just a string representation.

Example test cases:

| Input | Expected Output | Actual Output | Pass ? |
|--|--|--|--------|
| Constructor: numStrings=7 guitarLength=30.2 Manufacturer=Fender Color=Black | ** Output ** toString(): (numStrings=7, Length=30.2, manufacturer=Fender, color=Black) getNumStrings(): 7 getGuitarLength(): 30.2 getGuitarManufacturer(): : Fender getGuitarColor(): Black playGuitar(): [A(2), G(3), B(0.5), C(1), C(1), D(0.25), E(2), F(2), G(0.25), C(4), C(1), F(0.25),A(1), C(2), D(4),C(4)] | ** Output ** toString(): (numStrings=7, Length=30.2, manufacturer=Fender, color=Black) getNumStrings(): 7 getGuitarLength(): 30.2 getGuitarManufacturer(): : Fender getGuitarColor(): Black playGuitar(): [A(2), G(3), B(0.5), C(1), C(1), D(0.25), E(2), F(2), G(0.25), C(4), C(1), F(0.25),A(1), C(2), D(4),C(4)] | Yes |
| Test case 2 here | | | |
| Test case 3 here | | | |

The google recommended Java style guide, provided as link in the week 2 content, should be used to format and document your code. Specifically, the following style guide attributes should be addressed:

- Header comments include filename, author, date and brief purpose of the program.
- In-line comments used to describe major functionality of the code.
- Meaningful variable names and prompts applied.
- Class names are written in UpperCamelCase.
- Variable names are written in lowerCamelCase.
- Constant names are in written in All Capitals.
- Braces use K&R style.

Submission requirements

Deliverables include all Java files (.java) and a single word (or PDF) document. The Java files should be named appropriately for your applications. The word (or PDF) document should include screen captures showing the successful compiling and running of each of the test cases. Each screen capture should be

properly labeled clearly indicated what the screen capture represents. The test cases table should be included in your word or PDF document and properly labeled as well.

Submit your files to the Homework 2 assignment area no later than the due date listed in your LEO classroom. You should include your name and HW2 in your word (or PDF) file submitted (e.g. firstnamelastnamehw2.docx or firstnamelastnamehw2.pdf)

Grading Rubric:

The following grading rubric will be used to determine your grade:

| Attribute | Meets |
|-------------------|--|
| Guitar Class | 10 points A private int data field named numStrings that defines the number of strings on the guitar was included. The default value was 6. (0.5 point) A private double data field named guitarLength that defines the length of the guitar in inches was included. The default value was 28.2. (0.5 point) A private String data field named guitarManufacturer that defines the manufacturer of the guitar was included. The default value was "Gibson". (0.5 point) A private Color data field named guitarColor that defines the color of the guitar was included. The default value was Red. (0.5 point) A no argument constructor that creates a Guitar using the default number of strings, length, manufacturer and color was included. (1 point) A constructor that creates a Guitar using a specified number of strings, length, manufacturer and color was included. (1 point) Getter methods were included for all data fields. (2 points) A playGuitar() method that returns a string representation of 16 randomly selected musical notes of random duration was included. (2 points) A toString() method that displays the number of strings, length, manufacturer and color in String format was included. (2 point) |
| Test Guitar Class | 5 points |

| | |
|-------------------------------|--|
| | <p>TestGuitar class was used to construct at least, different 3 Guitar objects. (3 points)</p> <p>For each of the objects constructed, each of the methods was called with results output. (2 point)</p> |
| Test Cases | <p>5 points</p> <p>A minimum of 3 test cases was used in the form of table with columns indicating the input values, expected output, actual output and if the test case passed or failed. The table should contain 4 columns with appropriate labels and a row for each test case. (5 points)</p> |
| Documentation and Style guide | <p>5 points</p> <p>Screen captures were provided and labeled for compiling your code, and running each of your 3 test cases. (1.5 points)</p> <p>Header comments include filename, author, date and brief purpose of the program. (0.5 points)</p> <p>In-line comments used to describe major functionality of the code. (0.5 points)</p> <p>Meaningful variable names and prompts applied. (0.5 points)</p> <p>Class names are written in UpperCamelCase. (0.5 points)</p> <p>Variable names are written in lowerCamelCase. (0.5 points)</p> <p>Constant names are in written in All Capitals. (0.5 points)</p> <p>Braces use K&R style. (0.5 points)</p> |