**Programming 2 – Spring 2021**
**MP1 Scoresheet – 9 February 2021**

Student Name: _____
**(must be filled out)**
Lab section: morning afternoon

**All MP1 source code (.java) files and Screenshots are to be turned in by the end of day Tuesday, 9 February.**

1. Adherence to specifications (5 point **DEDUCTION** for each violation):

| Method present? | Method used as specified? | Method and signature |
|---|---|---|
| **YES** \| no | **YES** \| no | `RandomMooValue()` |
| **YES** \| no | **YES** \| no | `int getBigMooCount( int guess )` |
| **YES** \| no | **YES** \| no | `int getLittleMooCount( int guess )` |
| **YES** \| no | **YES** \| no | `int getSecretValue()` |
| **YES** \| no | **YES** \| no | `boolean isCorrectGuess( int guess )` |
| **YES** \| no | **YES** \| no | `boolean setSecretValue()` |
| **YES** \| no | **YES** \| no | `boolean setSecretValue( int n )` |

2. Demonstrate the following runs of your program via Screenshots. 10 points per run.
5-point deduction for each observed error within a run. 10-point deduction for missing runs of your program.

At start: answer is hidden from the user within the GUI. ( yes – 5 points ) ( no )

| **Run #1:** | 0000 | 1111 | 2222 | 3333 | 4444 | 5555 | 6666 | 7777 | 8888 | 9999 | Points: _____ / 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # of MOO! | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | |
| # of moo. | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | |

"All you hear are cowbells" message displayed when there's nothing being "uddered"… ( yes – 5 points) ( no )
Terminates after 10 incorrect guesses? ( yes – 5 points) ( no )
Message dialog box used to display end game result. ( yes – 5 points) ( no )
Program displays "Boo hoo – no LaurieMOO." and displays the secret four-digit value at end. ( yes – 5 points) ( no )

| **Run #2:** | 0011 | 2233 | 4455 | 6677 | 8899 | 1234 | 5678 | 9090 | Points: _____ / 10 |
|---|---|---|---|---|---|---|---|---|---|
| # of MOO! | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | |
| # of moo. | ____ | ____ | ____ | ____ | ____ | ____ | ____ | ____ | |

Program displays appropriate combinations of "MOO!" and/or "moo." as needed;
Location of the digits are NOT indicated! ( yes – 5 points) ( no )

List deductions here:

---

**Testing/Validation Subtotal:** _____ **/ 50**

**Documentation Subtotal:** _____ **/ 35**

**Variables Subtotal:** _____ **/ 15**

**DEDUCTIONS:** < _____ >

**TOTAL:** _____ **/ 100**

**Programming 2 – Spring 2021** Student Name: _____
**MP1 Scoresheet – 9 February 2020**
**Source Code Analysis Rubric** Lab section: morning afternoon

**Readability – 20 points total**

| Criteria | Meets Expectations - 5 | Needs Minor/Major Improvement – 4/3/2 | Unacceptable/Missing - 0 |
|---|---|---|---|
| Organization<br><br>Score (x1): _____ | Code is broken down into clear, recognizable, well thought out sections of functional units; blank lines and comments used to establish visual structure. | | |
| Separation<br><br>Score (x1): _____ | Spaces used as appropriate to help differentiate distinct elements within each coding statement. | | |
| Consistency<br><br>Score (x1): _____ | Similar coding constructs regularly use the same format regarding indentation and alignment; similar or related variable names follow an established pattern. | | |
| Grammar<br><br>Score (x1): _____ | All comments employ proper sentence structure, capitalization, word choice, and punctuation. No spelling errors noted. | | |

**Documentation – 30 points total**

| Criteria | Meets Expectations - 5 | Needs Minor/Major Improvement – 4/3/2 | Unacceptable/Missing - 0 |
|---|---|---|---|
| Class Header Comments<br><br>Score (x1): _____ | Every class file starts with a header comment that contains the name of the file, the date of its writing, the full name of its author, and a description of what the class does. | | |
| Method Header Comments<br><br>Score (x2): _____ | Description clearly but succinctly explains purpose of the function. Preconditions and postconditions as specified as appropriate. | | |
| Javadoc comments<br><br>Score (x1): _____ | Javadoc appropriately used with all public methods for the classes within the project. | | |
| Section Comments<br><br>Score (x1): _____ | Each functional section of code includes a comment describing the goal or purpose that that section is trying to accomplish without being either verbose or parroting. | | |
| Code Comments<br><br>Score (x1): _____ | Line-oriented comments are used to clarify meaning and/or provide elaboration as needed. | | |