Grade: Excellent job! 10/10. Please see comments below.

If one of the following is true, you will NOT get credits.

☐The project is late.
☐ The algorithm or class design is missing.
☐The project has errors.
☐ There are no comments (Javadoc format required) at all in your code.
☐Wrong files are submitted.
☐ A project is a copy and modification of another student's project. (Both will receive 0.)

Software Development Project Rubric: Design is worth 20%; the rest is worth 80%.
Analysis
Note: There will be no credit if the software doesn't meet customer specification at all.
Does the software meet the exact customer specification?
Does the software read the exact input data and display the exact output data as they are shown in sample runs?
Does each class include all corresponding data and functionalities?

Design
Note: There will be no credit if the design is missing.
Is the design (a UML class diagram) an efficient solution?
Is the design created correctly?

Code
Note: There will be no credit if there are syntactic errors.
Are there errors in the software?
Are code conventions and name conventions followed?
Does the software use the minimum computer resource (computer memory and processing time)?
Is the software reusable?

This is a way of shuffling cards and calculating card values:

1. Store numbers from 1 to 52.

```
ArrayList<Integer> number1To52 = new ArrayList<Integer>();
for(int i = 0; i < 52; i++){
   number1To52.add(i+1);
}
```

2. Then, you shuffle the 52 numbers like this:

```
Collections.shuffle(number1To52);
```

3. Calculate the card values like this:

```
int cardValue1 = (numbers.get(0) % 13;
int cardValue2 = (numbers.get(1) % 13;
int cardValue3 = (numbers.get(2) % 13;
int cardValue4 = (numbers.get(3) % 13;
```

Debug

Are there bugs in the software?

Documentation

Note: There will be no credit if comments are not included.

Are there enough comments included in the software?

Class comments must be included before a class header.

Method comments must be included before a method header.

More comments must be included inside each method.

All comments must be written in Javadoc format.