

Name _____ TA _____ Login name _____

- Turn off your cell phone to avoid losing a letter grade when it rings.
- Use the amount of space provided to gauge how much you should write. Brevity is the soul of points. Points are not related to wit.
- Legibility counts, so be neat. If your writing is smaller than the typeface of this exam, I may deduct points
- Points may be deducted for irrelevant, meaningless, or contradictory statements (and of course, just plain false statements). Please be sure to answer the question I asked!
- Do not complicate an example. Do not make up features of an example unless directed to do so. Simple is best!
- Some questions look hard at first, but if you `for(i=0;i<3;i++) breatheDeeply();` you realize it is simpler than you first thought.
- **Do not change code I have written** unless explicitly directed to do so.

1. (6 pts) A scrum is a _____ that lasts about _____ with a frequency of _____.
2. (6 pts) In Git, what is the difference between push and commit?
3. (6 pts) Which design pattern can be modeled by the Swing implementation of JButton?

4. (6 pts) When a Set s needs to check whether an object is a duplicate, first s calls _____ because it is _____, and then s calls _____.
5. (6 pts) When we pass a Comparator object as a parameter, instead of passing a method, that is an example of _____.
6. (6 pts) Java has interfaces Comparable and Comparator. As discussed in class, what is the main advantage of Comparator?
7. (6 pts) Who is invited to a sprint review?
8. (6 pts) What is the output of a sprint retrospective?
9. (6 pts) Scrum is a form of _____ methodology,
which gives a lot of power to _____.
The older methodology we discussed briefly in class was _____.
10. (6 pts) The _____ is responsible for estimating how long it will take to complete a task.
11. (6 pts) The duration of a scrum is about _____.
12. (6 pts) Will the second midterm potentially contain material from the first midterm and every class exercise?
(Hint: true)

```
//Write minimal code. You may not need all lines.  
//A FriedEggException is a kind of Exception.
```

Define a minimal FriedEggException to be used below.

```
class TryCatch{
```

```
    //Write a very short method that will always raise a FriedEggException.
```

```
    public static void main(String[] args){
```

```
    //Call the method you defined and write code to handle the exception by  
        //printing it (or the stack).
```

 }
}

(20 pts) Assume two cows are the same if they have the same name. Write class Cow and the necessary methods to make main this work, and put a print statement inside each method. Then on the lines next to each call to add, write what the methods would print.

```
class HashSetPractice {  
  
    /*  
     * On the comment lines provided, write *all* that is printed as a  
     * result of each line of code.  
     */  
  
    public static void main(String[] args) {  
        HashSet<Cow> herd = new HashSet<Cow>();  
  
        herd.add(new Cow("A"));           // _____  
        System.out.println();  
        herd.add(new Cow("Alba"));        // _____  
        System.out.println();  
        herd.add(new Cow("A"));           // _____  
        System.out.println();  
        herd.add(new Cow("Aga"));         // _____  
        System.out.println();  
        herd.add(new Cow("A"));           // _____  
        System.out.println();  
        System.out.println(herd.size()); // _____  
  
    }  
}
```