Set 08 Study Questions

1.   Consider the following class:

```
public class Tornado {
    private static final int MAX_STRENGTH = 7;
    public String name = _Bill_;

    private void causeDamage() {
        // imagine code here
    }

    public static int calculateTrajectory() {
        // imagine code here
    }
}
```

Below, assume that the variable _t_ refers to a Tornado object. The table below has rows corresponding to various expressions related to the Tornado class. The columns list various places where one might consider using those expressions. Clearly place a _V_ in any box where the corresponding expression is *valid syntax* when used in the corresponding context.

	static method of Tornado class	non-static method of Tornado class	method of some other class
Name			
MAX_STRENGTH			
causeDamage()			
calculateTrajectory()			
t.name			
t.MAX_STRENGTH			
t.causeDamage()			
t.calculateTrajectory()			
Tornado.name			
Tornado.MAX_STRENGTH			
Tornado.causeDamage()			
Tornado.calculateTrajectory()			

2. Where can you use a _continue_ statement?

3. Describe how a continue statement behaves in a while loop.

4. Describe how a continue statement works in a for-loop.

5. Where can you use a _break_ statement?

6. Describe how a break statement behaves.

7. Write some faulty code that generates a null-pointer exception, catch the exception immediately and print out _exception caught_ in your catch block.

8. Write some code that prompts the user for a numerical value, and reads their input into an `int` variable. Run the program, and try entering some text (like _cat_) instead of a number. Notice what kind of exception is thrown. Now modify your program so that it catches this exception, and instead of crashing the program, have it tell the user that he/she must enter a NUMBER, and then prompt them for input again.

9. Write a method called `smallSum` that takes two int parameters, `x` and `y`. If the absolute value of the sum of the integers is more than 100, throw an ArithmeticException, passing the String _I don_t like big numbers_ to the constructor of the exception. If the sum is less than 100, then return the sum. Write a quick driver to test out your method. After making sure everything works correctly, modify the driver so that it catches the exception and prints out the message that was passed to the exception_s constructor, but doesn_t crash the program.

10. Explain the relationship between exception handling and the call stack. What happens if an exception is thrown but not caught anywhere in your program?

11. Under what circumstances will the finally block run?