**Pencil and Paper Assignment for Lesson 7**

1. **(d) StackOverflowError**

***Note****: RandomNumbers.getRandomInt() is not a built-in method in Java. So, I imported the package from Lab2. Otherwise, the code will show* ***Compiler Error***

**Explanation**: The main method is creating an instance of the class and calling the constructor of the class. And the **MyClass** constructor calls the recurse method with the argument String "Hello". The recurse method recursively calls itself with a substring of the input string, either the first half or the second half, depending on the result of random number is even or odd.

The recursion will continue until the substring is of length 0, and at a specific point s.substring will return empty strings, causing an infinite recursion of empty string concatenation. This will throw a **StackOverflowError** as a result of bad recursion.

1. **(d) StackOverflowError**

**Explanation**: The main method is creating an instance of the class MyClass and will call the constructor of the class. The class constructor will run the method recurse with the string parameter “Hello”. It will permute (or rearrange) the string and generate a string with same length. Again, the recurse method will be called with the newly generated string parameter. As this is an unstoppable call of recursion, the program will throw a **StackOverflowError**

1. To force the code to use the clone() method of Enclosing instead of the one in object, we need to call the clone() method to the Enclosing class explicitly. (The modification marked in yellow)

**package** assignments.lab7.pencil;

**public** **class** Enclosing **implements** Cloneable {

**public** Enclosing clone() **throws** CloneNotSupportedException {

System.***out***.println("Inside Enclosing.clone()");

**return** (Enclosing) **super**.clone();

}

**class** Inner **implements** Cloneable {

**void** innerMethod() **throws** CloneNotSupportedException {

Object copy = Enclosing.**this**.clone();

System.***out***.println(copy.getClass().getName());

}

}

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

Enclosing p1 = **new** Enclosing();

Enclosing.Inner i1 = p1.**new** Inner();

**try** {

i1.innerMethod();

}

**catch** (CloneNotSupportedException e) {

e.printStackTrace();

}

}

}