1. The CreateNewOp() method in DrawingUndoStackSimulation has a return type of DrawingOp. This method randomly creates and returns one of 8 different operations. In cases 4-7, a ColorOp object is created and added to the myOpStack, even though it is not a DrawingOp. How is this possible? **The ColorOp object is a subclass of DrawingOp. Therefore it can be used.**

2. Why do we use more than one case in this switch statement for pushing new drawing

objects? (If you aren’t sure, comment out case 0 and case 1, leaving only case 3 to

push new operations, and run the project to see what’s different). **It allows for no matter what number you generate. It will always perform an operation. This is because case zero in case one use something called fall through, which means that it will continue until it finds another case statement that will have a break statement inside of it.**

3. What happens if you comment out the try-catch structure that surrounds the pop and

peek operation? Try it and see – run the simulation multiple times until you see the

problem in the output. **This is because if you look in the documentation of .pop(), it will actually throw an empty collection exception the list is empty, meaning, you will need a try statement when you use it.**

4.What happens without the try-catch structure? Why does this

happen? **It will give you a error that says a try, catch statement is needed, this is because things could probably go very wrong if you don’t have one.**

5. Go to the printStack method. Above the for loop, type `myOpStack’, then type a period

and see what methods NetBeans offers you for this stack. You should see something like

this:There are many methods here, because Java’s stack implementation inherits from

another collection, Vector. If you scroll through this list, what do you notice about the

operations that are specific to a Stack (such as push and pop)? **They are in bold font**

6. What methods are available for Java stack objects that really shouldn’t be, given how a

stack operates and what it is designed to do? **getIndexOf(), clear() , clone(), search()**