1. **Did the throws InputMismatchException clause cause any errors? Why, or why not?**

It did cause an error that stated,” InputMismatchException cannot be resolved to a type”. This was because InputMismatchException needed to be imported.

1. **Describe what happened? Include your interpretation of the output displayed. Where is the exception object created?**

An InputMismatchException was thrown. The output displayed shows exactly where the exception was created. The exception object was created in the main method.

1. **Describe the output displayed. How is it different than the output displayed previously?**

**Describe what happened this time vs. what happened previously. Please be specific. What was the last statement to execute within the try block?**

The output displayed comes from the try catch block. The exception is caught so therefore no exceptions are thrown. The designated outcome is produced. The last time an exception was thrown because there was no try catch block. The last statement to execute was the parameter from the try catch block.

1. **Explain why the statement done = true; is placed at the end of the try block. Was there a difference in output when you added the nextline statement to the catch block?**

The statement done = true is placed at the end of the try block because we don’t want to break the loop until we have valid input. There was not a difference when we added to nextLine statement.

1. **Why must you enter all three values (month, day, and year) before your program determines that month is invalid?**

The object is not instantiated until all of the values have been entered. The instantiation occurs after we enter all three values.

1. **What error do you get? Move the catch block/handler so that it is below the others. You shouldn’t get the error now. Why do you think the error was produced when the handler appeared first?**

The error we get is that InputMismatchException is already handled by the catch block for RuntimeException. The error was produced because the exception was already being handled since the runtime catch came first. The result of switching the two catch blocks is that the InputMismatchException is not being handled yet.

1. **Describe what happens and the output displayed. How is it different than the output displayed in step #13?**

The exception is caught so it is no longer thrown. The month is out of range therefore the user is directed to re-enter a valid month. The output in step 13 was an exception that was thrown but not caught.

1. **Unreported exception errors are produced. Why? Please be specific. Include the code required to fix these errors. Please ask your instructor if you do not know what code is required.**

These errors are produced because we did not include the statements to throw DateException at the end of the method headers of the setMonth and Date parameterized constructor.

1. **Recall that the setMonth method previously threw a RuntimeException and did not need to have a throws clause appended to the end of the method header. Why does it need a throws clause now?**

The RuntimeException is an unchecked exception therefore it does not need to be checked. The DateException is a checked exception and it need to be thrown in the method header.

1. **Describe the code that is executed after the user enters 13, 12, and 2016. Be specific. What type of exception object is created? Where is the exception object created? Which statement in the try block is the last to be executed?**

DateException: Invalid Month: month out of range. A DateException object is created. It is created once we call the setMonth method. The last statement to be executed is done = true.

1. **What is the difference between the keyword “throw” and the keyword “throws”?**

The difference between “throw” and “throws” is that “throw” is used to throw an exception. The word “throws” is used in a method or class header to declare the possibility that a certain exception can occur. Once declares it and the other does it.