1. The throws InputMismatchException clause did cause an error. The error stated that InputMismatchException cannot be resolved to a type. The reason for this is because I had to import java.util.InputMismatchException; After I imported that line the error went away and the program was able to run.
2. After I entered ‘w’ for the month I got a InputMismatchException. The reason for this is because the input was not an integer. The output displayed tells you the type of exception as well as where it occurred. The exception object is created in the method that it occurs.
3. This time upon running the program you get a message stating Invalid input entered. Enter an integer. Then you can enter a valid integer. This is different from the previous output because this time an exception was not thrown. This time the exception was caught and allowed you to enter valid input. The last time an exception was thrown and the program stopped running. The last statement to execute within the try block was: month = scan.nextInt();
4. The reason that done = true; is placed at the end of the try block because all of the desired lines have been executed and we do not need to loop through anymore. The difference in output after adding the nextLine statement is that if you enter something other than an integer the program will loop through and ask you to enter the information again. Previously it would be stuck in an endless loop.
5. The reason you have to enter all three parts of the date is because the setMonth method is not called in the constructor until all three parts of the data have been entered.
6. The error that you get is Unreachable catch block for InputMismatchException. It is already handled by the catch block for RuntimeException. The error was produced before because as it states it is already handled by the RuntimeException catch block. The error went away after being moved because it changes the order of execution therefore it is not being handled by the RuntimeException first.
7. The program catches the exception and displays an error message. It then allows you to enter a month again. It is different from the output displayed in step 13 because in step 13 the exception was not caught.
8. The two unreported exceptions errors are Unhandled exception type DateException. These are produced because the exceptions are unhandled. It shows up after being thrown in the setMonth method because it is in the method header so therefore must be handled when used outside of the method.

**Code to Fix:**

**public** **Date**(**int** month, **int** day, **int** year) **throws** **DateException** {

setMonth(month);

setDay(day);

setYear(year);

}

**public** **void** **setDate**(**int** month, **int** day, **int** year) **throws** **DateException** {

setMonth(month);

setDay(day);

setYear(year);

}

1. The reason it now need a throws clause at the end is because RuntimeException is an unchecked exception. A DateException is a checked exception so it needs to be acknowledged.
2. After the user enters the desired parameters you get a handled DateException that tells you the month is out of range and then ask you to enter a month again. The exception object is created in the Lab3App class after the try block. The last statement to be executed in the try block is done = true;
3. The difference between the keyword “throw” and the keyword “throws” is that throw is used to throw our own exception in the program and throws is to declare an exception inside of a method.