

ACS-1904 W2024

Lab #9: Exception handling

Due by Friday, March 15 at 11:59 pm

- Compress the **entire BlueJ project folder** into a .zip file called Lab9.zip.
- Submit Lab9.zip containing the BlueJ project to the Nexus dropbox. Note that the data files and the extra work will be part of the .zip file as well, but don't worry about that.
- **Include your name and student number in each file as a comment.**

For this lab, you will need to download the BlueJ project from:

<https://github.com/rsveinson/ACS-1904-2024-Lab9.git>

- 1) Create a program called `ReadNamesFromXML` that reads the file `lab09.xml`, stores each name in the file into an `ArrayList` and then displays them by printing the `ArrayList`. The `ArrayList` should be completely loaded with the names before it is printed, don't load and print the list at the same time. The number of names in the file is not known.

On the documentation page for `XMLDecoder` you will find that exceptions can arise when this program runs:

- `FileNotFoundException`: when the `XMLDecoder` object is created, and the indicated file is not in the correct folder. This could also be caused by a typo in the name of the file in the code.
- `ArrayIndexOutOfBoundsException`: when the input stream has no more objects. For this exception, you need the import:
`java.lang.ArrayIndexOutOfBoundsException.`

Requirements:

- a. The program should terminate with the message "Error, file not found" and "End of Program" if a `FileNotFoundException` occurs.

Note: don't use `System.exit()` or any other means to terminate the program early. The program should continue to the end of the main method without reading the contents of the file, printing the `ArrayList` or closing the decoder.

- b. Since you do not know how many names are in the file -- it should read as many names as the file holds.
 - Use `ArrayIndexOutOfBoundsException` to accomplish this

- 2) The Java class `CatsXML.java` is supposed to write some `Cat` objects to the XML file `cats.xml`. As you can see, if you open the file, the `Cat` objects are not being written. Fix the `Cat.java` class so that it meets all of the requirements to write to an XML file. Be careful here, there's more than one missing element in the class `Cat.java`.

- a) Now that you have fixed `Cat.java` so that the driver writes the `Cat` objects to the XML file add the necessary code so that it;
 - i) Opens the `XMLDecoder` stream to the `Cats.xml` file.
 - ii) Since `XMLDecoder` could throw a `FileNotFoundException` use a try-catch so that a "File Not Found" error prints to the terminal window if the file is not found. If the file is not found the list should not be loaded and the stream should not be closed.

- b) The `loadCats (...)` method will read `Cat` objects from the stream until it reaches the end of the file, i.e. an EOF-style loop. Remember that the `XMLDecoder` will throw an `ArrayIndexOutOfBoundsException` when there are no more objects in the stream.

Sample output 1: Successful stream and read from file.

```
Emerson 17
Rincewind 1
Alax 16

----- Print file contents -----
Reading XML file.
eof reached

Emerson 17
Rincewind 1
Alax 16
end of program
```

Sample output 2: `FileNotFoundException`

```
Emerson 17
Rincewind 1
Alax 16

----- Print file contents -----
File Not Found

end of program
```

Submit your java file (`ReadNamesFromXML.java`) via Nexus.

EXTRA WORK: this will not be graded

The program [TotalNumbers](#) prompts the user to enter numbers until the user signals they are finished with "q", then the system returns the sum of the numbers. A user can easily mistype an integer (i.e. pressing a non-numeric key, or entering a number with a decimal like 2.5) and the program would fail abruptly with a `NumberFormatException` (on the [documentation page for Integer](#), see `parseInt`).

Modify the program to handle such an error: inform the user that the input is invalid and prompt the user to re-enter the number.