CSC 122: Text Encryption

Exception Handling

1 Introduction

In this lab, students will be provided with a buggy text encryption application for android to reinforce the concept of exception and error handling techniques. This application will contain a text box in which the students can enter a text string. After pressing the Encrypt button, the text will be show in another box as an encrypted string. By pressing the Decrypt button, students will be presented with their original text. The students will have to debug this code and decide where to place try/catch statements in order for the application to work correctly.This lab will demonstrate the importance of error checking/handling in complex code.

2 Objective

The purpose of this lab is to teach the idea of exception handling. After fixing the buggy Encryption application, the students will learn how to use try, catch and throw blocks to handle exceptions.By working on this lab students will be able to write robust software program by learning techniques for exception handling.

3 Activity

This section describes the tasks to be completed that will satisfy the given objectives.

3.1 Background Information

An exception is an event, that occurs at the time of the execution of a program, which breaks the normal flow of the program's instructions. A program can use exceptions to point out that an error has taken place. To throw an exception, use the throw statement and provide it with an exception object (a descendant of Throwable) to give information about the specific error that occurred. A throws clause must be included in the decleration of method if a method throws an uncaught, checked exception. A program can catch exceptions by utilizing a combination of the try, catch, and finally blocks. The try block recognizes a block of code in which an exception can occur. The catch block determines a block of code, known as an exception handler, which can manage a particular kind of exception. The finally block identifies a block of code that will certainly execute, and is the appropriate place to close files, recover resources, and otherwise clean up after the code

confined in the try block. The try statement must have at least one catch block or a finally block and can have multiple catch blocks. The type of exception thrown can be determined by the class of the exception object. More information about the error can be found by the exception object. With exception chaining, an exception can direct to the exception that caused it, which can successively detail about the exception that caused it, and so forth.

3.2 Implementation

This section will guide you in completing the labs.

3.2.1 Add Throw Clauses

Use the throw statement for `generateKey()`, `encrypt()` and `decrypt()` methods to handle exceptions.

3.2.2 Add Try and Catch blocks

Use try and catch blocks to manage the exception for `generateKey()`, `encrypt()` and `decrypt()` methods. In the catch block use `printStackTrace()` method of throwable class to handle the exception by printing the stack trace to the System.Err stream.

4 Conclusion

After the successful completion of this lab the students will learn how to catch and handle exceptions. The students will use throw, try and catch blocks to catch different types of exception and provides a message if an exception is found.

5 Deliverables

To submit your application, export your Eclipse project as a file system, zip all of the files into an archive and submit them online with the filename `<first_name_initial><last_name>-lab<lab#>.zip`.