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.

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions. A program can use exceptions to indicate that an error occurred. To throw an exception, use the throw statement and provide it with an exception object (a descendant of Throwable) to provide information about the specific error that occurred. A method that throws an uncaught, checked exception must include a throws clause in its declaration. A program can catch exceptions by using a combination of the try, catch, and finally blocks. The try block identifies a block of code in which an exception can occur. The catch block identifies a block of code, known as an exception handler, that can handle a particular type of exception. The finally block identifies a block of code that is guaranteed to execute, and is the right place to close files, recover resources, and otherwise clean up after the code enclosed in the try block. The try statement should contain at least one catch block or a finally block and may have multiple catch blocks. The class of the exception object indicates the type of exception thrown. The exception object can contain further information about the error, including an error message. With exception chaining, an exception can point to the exception that caused it, which can in turn point to the exception that caused it, and so on.

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 try, catch and throw blocks to hadle exceptions.By working on this lab students will be able to write robust software program by learning techniques for exception handling.

3 Activity

1. Use throw clause for `generateKey()` method to handle exception.
2. Use try and catch blocks to handle the exception for `generateKey()` method. In the catch block use `printStackTrace()` method of throwable class to handle the exception.
3. Use throw clause for `encrypt()` method to handle exception.
4. Use try and catch blocks to handle the exception for `encrypt()` method. In the catch block use `printStackTrace()` method of throwable class to handle the exception.
5. Use throw clause for `decrypt()` method to handle exception.
6. Use try and catch blocks to handle the exception for `decrypt()` method. In the catch block use `printStackTrace()` method of throwable class to handle the exception.

4 Conclusion

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

5 Deliverables

The deliverables for this project are as follows:

- The Eclipse project export with your version of EncryptActivity.java
- Lab Report