Homework 1

Configuring your Lab Environment

Fix your first Java Data Input issues

Overview

In this homework you will install Java JDK 8 along with the Netbeans Full IDE bundle. The full bundle provides support for Java, C/C++ and PHP languages. Multiple languages will be used as we detect and mitigate software errors. Becoming comfortable with your programming environment will set the foundation for the rest of the assignments.

If you completed SDEV 325 (which is a prerequisite for this course) you should have already installed the complete Netbeans bundle and essentially the first part of this assignment.

After configuring your environment, you will need to analyze and fix some security issues in a simple Java application that uses a command line argument to enter a filename.

Assignment

Part I: Environment Set-up and Simple Hello, World

To successfully complete this assignment (and this course), you will need to configure your environment for testing multiple languages. We will use Java JDK and Netbeans to provide this support.

1. Download and install Java JDK 8 from the Oracle site. Note, you may have previously installed JDK 7 but this course requires JDK 8. The download is available here:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

You may also need to configure your environment path in Windows. Review the installation guides on the Oracle web sites for details on how to do this.

2. Download and install Netbeans full IDE. The Full IDE includes Java, Java EE, PHP, C/C++ and server tools. The download is currently available here:

https://netbeans.org/downloads/

3. Once installed, you should review the online documentation and become comfortable starting and running the Netbeans IDE.
4. Using Netbeans, create your own unique Hello, World application using the Netbeans IDE for Java. Be sure your code runs properly within the Netbeans environment. Note: part of your unique Hello, World application should include the date and time stamp when the application was run. This should be within your Hello, World code. You can add other unique features as you see fit.

Part II: Fix security issues in a simple Java application that uses command line arguments.

1. Download the source file from this week. Found as an attachment in the homework folder.
2. Create a new Java application in Netbeans and either copy and paste the code or import the existing source file. Note you may need to make package adjustments if you created a different package

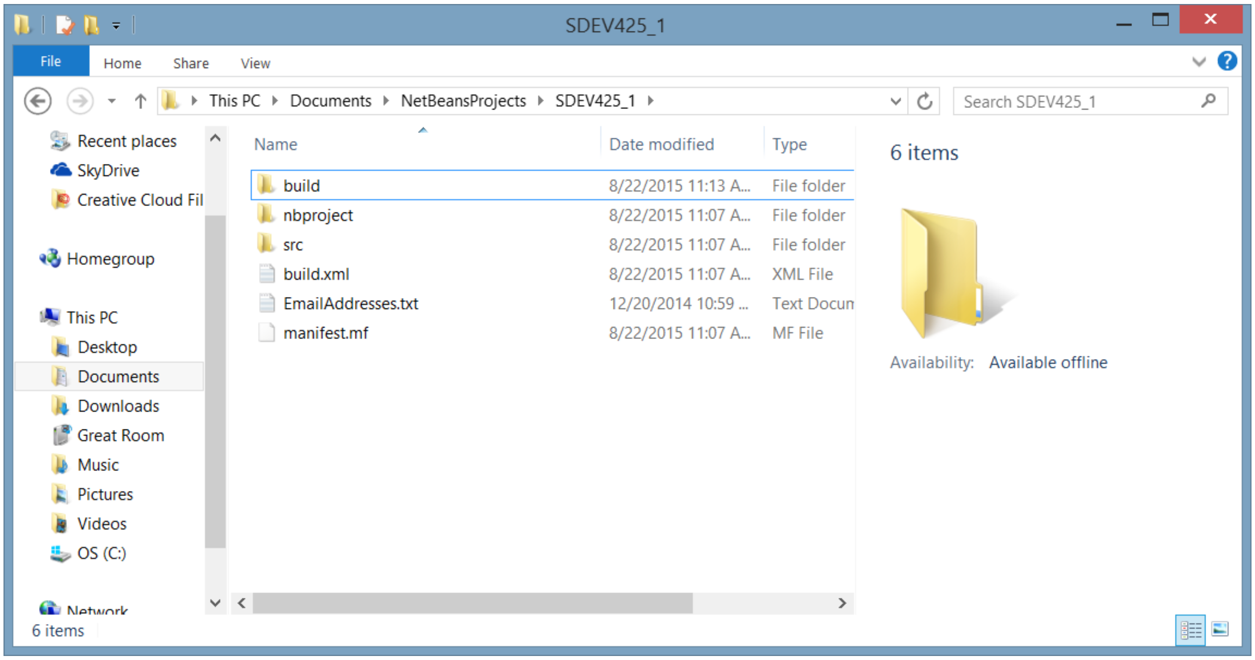
Here are text versions of the code and emailaddresses for your reference:

```
package sdev425;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

/**
 *
 * @author jim
 */
public class SDEV425_1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Read the filename from the command line argument
        String filename = args[0];
        BufferedReader inputStream = null;

        String fileLine;
        try {
            inputStream = new BufferedReader(new FileReader(filename));

            System.out.println("Email Addresses:");
            // Read one Line using BufferedReader
            while ((fileLine = inputStream.readLine()) != null) {
                System.out.println(fileLine);
            }
        } catch (IOException io) {
            System.out.println("File IO exception" + io.getMessage());
        } finally {
            // Need another catch for closing
            // the streams
            try {
                if (inputStream != null) {
                    inputStream.close();
                }
            } catch (IOException io) {
                System.out.println("Issue closing the Files" +
io.getMessage());
            }

        }
    }
}
```

EmailAddresses.txt

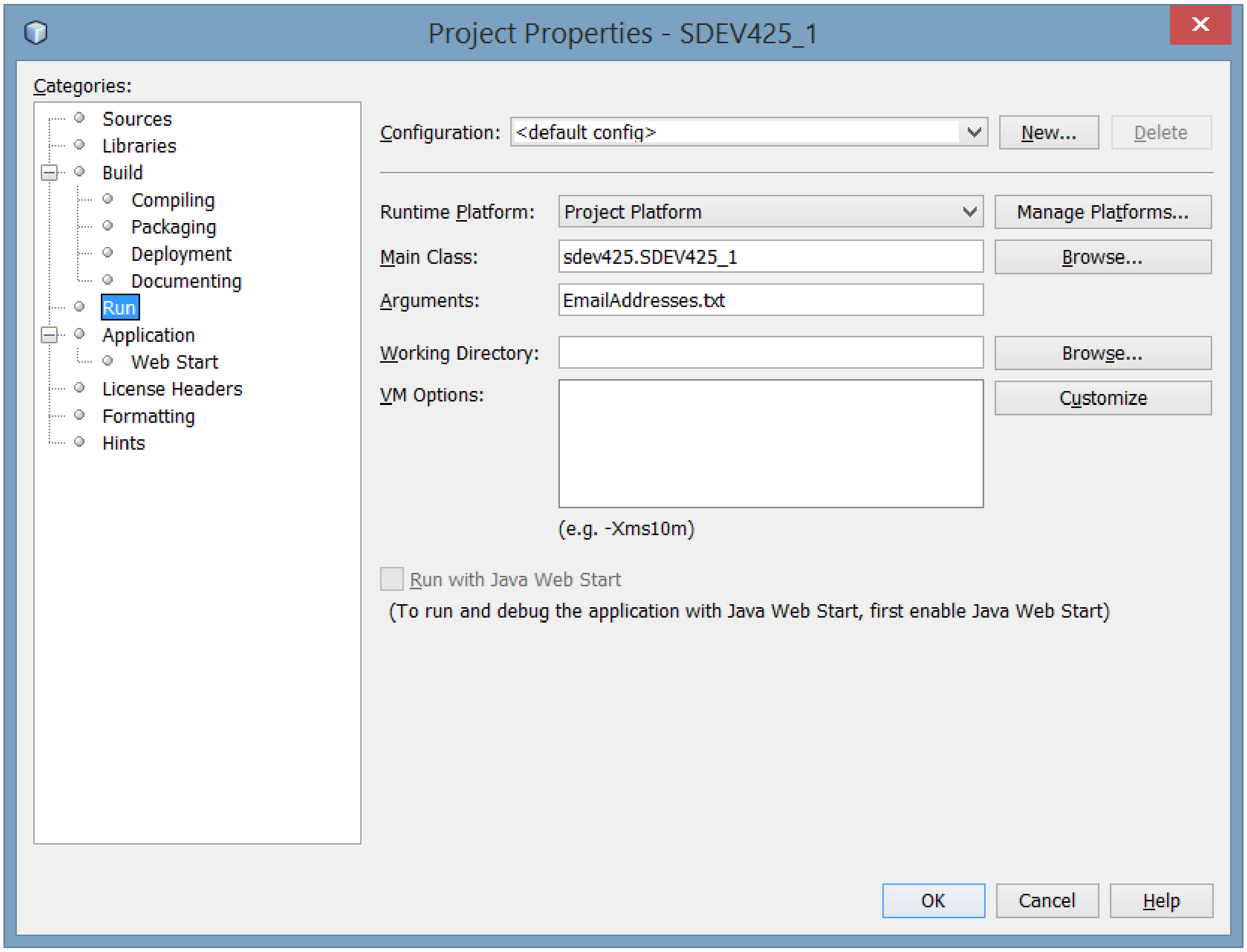
```
john@umgc.edu
fred@umgc.edu
susan@umgc.edu
donna@umgc.edu
javier@umgc.edu
jessie@umgc.edu
laura@umgc.edu
tina@umgc.edu
todd@umgc.edu
ed@umgc.edu
```

3. Be sure you place the EmailAaddresses.txt file in the Project folder in the root location:



Name	Date modified	Type
build	8/22/2015 11:13 A...	File folder
nbproject	8/22/2015 11:07 A...	File folder
src	8/22/2015 11:07 A...	File folder
build.xml	8/22/2015 11:07 A...	XML File
EmailAddresses.txt	12/20/2014 10:59 ...	Text Docun
manifest.mf	8/22/2015 11:07 A...	MF File

4. You can place the command line argument using the run properties of Netbeans. This screen is invoked by right clicking the project name and the selecting properties. Click on Run and enter the filename. This essentially simulates running java sdev425.SDEV425_1 EmailAddresses.txt at the command line prompt.



Configuration: <default config>

Runtime Platform: Project Platform

Main Class: sdev425.SDEV425_1

Arguments: EmailAddresses.txt

Working Directory:

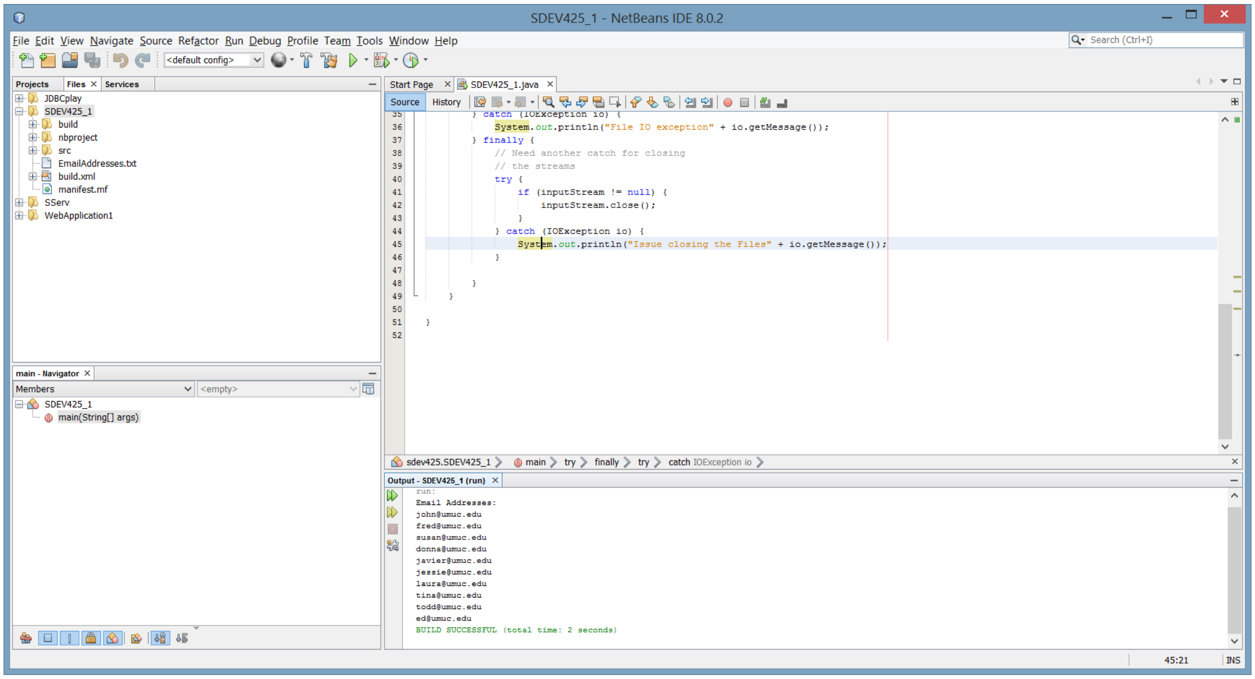
VM Options:

(e.g. -Xms10m)

□ Run with Java Web Start

(To run and debug the application with Java Web Start, first enable Java Web Start)

5. When you the run the application by clicking the green arrow, you should see output similar to this:



```
            } catch (IOException io) {
                System.out.println("File IO exception" + io.getMessage());
            } finally {
                // Need another catch for closing
                // the streams
                try {
                    if (inputStream != null) {
                        inputStream.close();
                    }
                } catch (IOException io) {
                    System.out.println("Issue closing the Files" + io.getMessage());
                }

            }
        }

}

```

```
run:
Email Addresses:
john@umuc.edu
fred@umuc.edu
susan@umuc.edu
donna@umuc.edu
javier@umuc.edu
jessie@umuc.edu
laura@umuc.edu
tina@umuc.edu
todd@umuc.edu
ed@umuc.edu
BUILD SUCCESSFUL (total time: 2 seconds)
```

6. Based on the reading and other research you perform, fix any security issues you find in this code.

Deliverables

Provide your security fixed Java source code along with a PDF document describing the issues you found, the rules that were being broken, the recommendations you applied, and specifically how you fixed it. Demonstrate, using multiple possible error or invalid input, the code works properly.

Your code should be well-documented with comments, include header comments, use proper variable and naming conventions and properly formatted. (Use the Google Java style guidelines provided in the content-> Course Resources). Be sure your PDF document is neat, well-organized and is well-written with minimal spelling and grammar errors. All references used should be included in your document.

Grading rubric:

Attribute	**Meets**	**Does not meet**
Simple Hello, World	**20 points** Creates a unique Hello, World application using the Netbeans IDE for Java. (5 points) Includes the date and time stamp when the application was run. (5 points) Demonstrates your code runs properly within the Netbeans environment. (10 points)	**0 points** Does not create a unique Hello, World application using the Netbeans IDE for Java. Does not include the date and time stamp when the application was run. Does not demonstrate your code runs properly within the Netbeans environment.
Fixes security issues	**60 points** Fixes any security issues you find in the provided code. (20 points) Describes the rules that were being broken. (10 points) Provides recommendations you applied and specifically how you fixed the code. (10 points) Demonstrates, using multiple possible error or invalid input, the code now works properly. (20 points)	**0 points** Does not fix any security issues in the provided code. Does not describe the rules that were being broken. Does not provide recommendations you applied and specifically how you fixed the code. Does not demonstrate, using multiple possible error or invalid input, the code now works properly.
Documentation and Submission	**20 points**	**0 points**

	Code is well-documented with comments, including header comments, use of proper variable and naming conventions and properly formatted. (5 points) Document is neat, well-organized and is well-written with minimal spelling and grammar errors. (10 points) All references used should be included in your document. (5 points)	Code is not well-documented with comments, including header comments, use of proper variable and naming conventions and properly formatted. Document is not neat, well-organized or well-written with minimal spelling and grammar errors. References were not included.