

**22/SP-COP-2805C-61163 Java Advanced**

**Assignment 12.15**

Document Version: 0.1

Version Date: 05/05/2022

Created By: Tristan Rogers

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Rationale |
| 0.1 | 2022 MAY 05 | Tristan Rogers | First Draft |

# Document Purpose

The purpose of this document is to outline and demonstrate the ability to read and write from a generated text file within Java.

# Technical Specifications

## Purpose of Technical Implementation

The purpose of this program is to demonstrate the ability to read and write to a file. The program will check to see if the file exists. If it does, it will write to the file. If it does not exist, the program will create the new file with the specified name and proceed to write to it. Using an ArrayList, the values from the file are then sorted and displayed.

## Technical Implementation Components

* This program uses a defined string as the filepath for the file that it will write t. It first checks to see if the file exists, and if not, it will create the file with the specified file name. a new FileWriter is used to input values into the file. The values are all integers created randomly through the Random() method from java.util. There are try catch block s implemented to handle IO and FileNotFound exceptions. Without them, the program will not work.

## Technical Implementation Pseudocode

*Search for file*

*if file exists*

*write to file*

*close file*

*else*

*create new file*

*write to file*

*close file*

*read file and store values in ArrayList*

*Sort ArrayList in increasing order*

*Display ArrayList results*

## Assignment 12-15 Implementation

This program contains only one main method. Variables are declared at the beginning of the method for ease of coding.

The first for loop is used to create 100 random integers to write into the file. Within this for loop is a try catch block to check if the file exists. If it does, the program will write to the existing file. If it does not, the system will create the file with the specified name and proceed to write to the file. The file is closed after these operations. The catch on this block will display an error message if there is an IOException.

An ArrayList is then created to store and sort the values to prepare them to be displayed. The try block uses a scanner to add values from the file into the ArrayList. A while loop continues this until there are no remaining integers. The catch block will activate if the FileNotFoundException is thrown.

The list is then sorted and displayed in increasing order.