Criterion C: Development

Contents

[Complex Data Types 2](#_Toc122345618)

[ArrayLists – 2D 2](#_Toc122345619)

[LinkedHashMap 2](#_Toc122345620)

[ArrayLists 3](#_Toc122345621)

[File Access 4](#_Toc122345622)

[Writing to File 4](#_Toc122345623)

[Write to Text File 4](#_Toc122345624)

[Write to Excel File 5](#_Toc122345625)

[Parsing Data 6](#_Toc122345626)

[Text File 6](#_Toc122345627)

[PDF 6](#_Toc122345628)

[Error Handling 7](#_Toc122345629)

[Try Catch 7](#_Toc122345630)

[External Libraries 7](#_Toc122345631)

[Apache Poi 8](#_Toc122345632)

[Apache PDFBox 8](#_Toc122345633)

[Javax 8](#_Toc122345634)

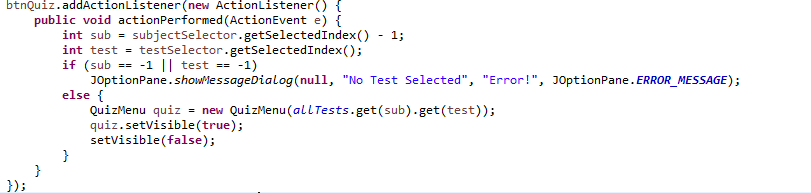
[Encapsulation 8](#_Toc122345635)

[Inheritance 8](#_Toc122345636)

Complex Data Types

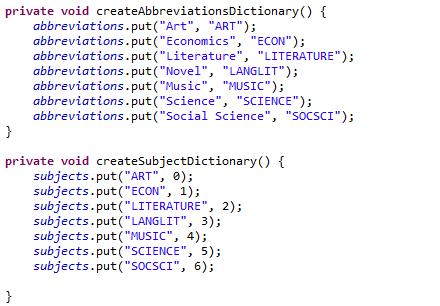
## ArrayLists – 2D

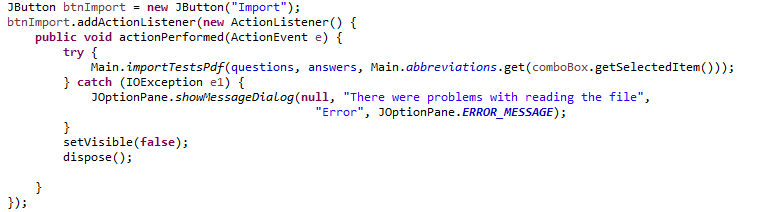




A two-dimensional ArrayList is utilized in the main class to easily store each test for each subject. This allows the code to easily access the data as it keeps the different subjects split. Any individual test could be found at any time by using the get() method twice with the subject’s index and then the test’s index. Therefore, when the user wants to import data and access the data later, it is much easier to accomplish these goals using a 2D ArrayList.

## LinkedHashMap

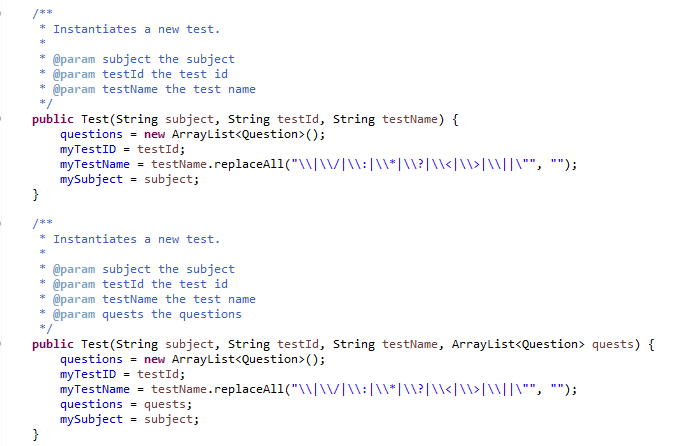
  




LinkedHashMaps are used to easily and quickly access and convert from associated data. Since each subject directly corresponds to a given index in the 2D ArrayList, LinkedHashMap was specifically chosen for this program because of the need to maintain order of the keys.

## ArrayLists





ArrayLists are used in the Test class to easily add questions to a specific test. Since there are not a set number of questions for each imported file, ArrayLists are the best solution as it does not have a defined length. Moreover, it also allows for easy access and modification of the data through the get() and set() methods.

# File Access

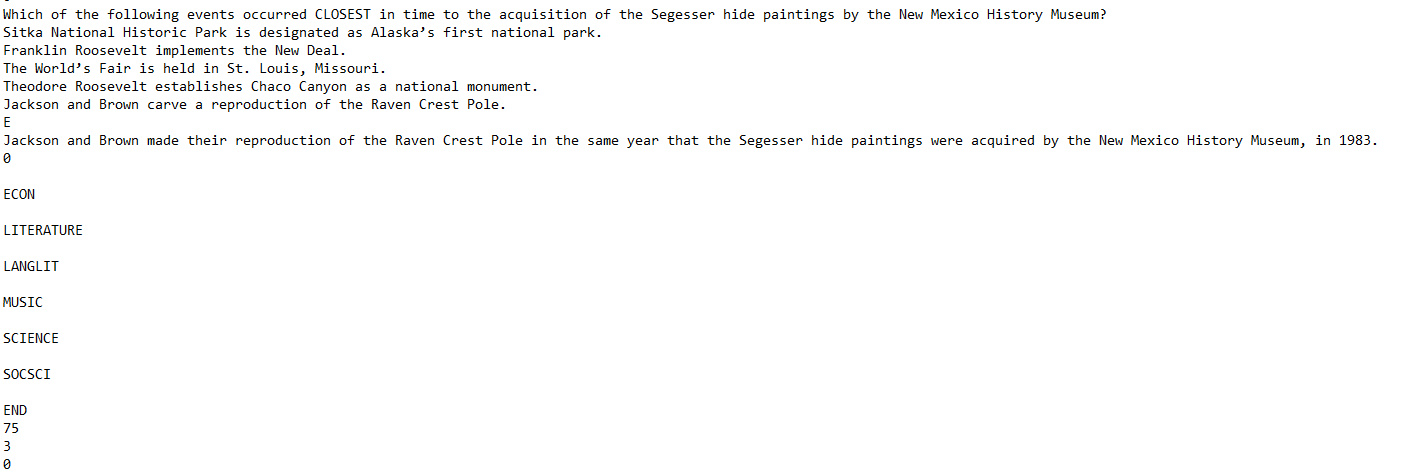
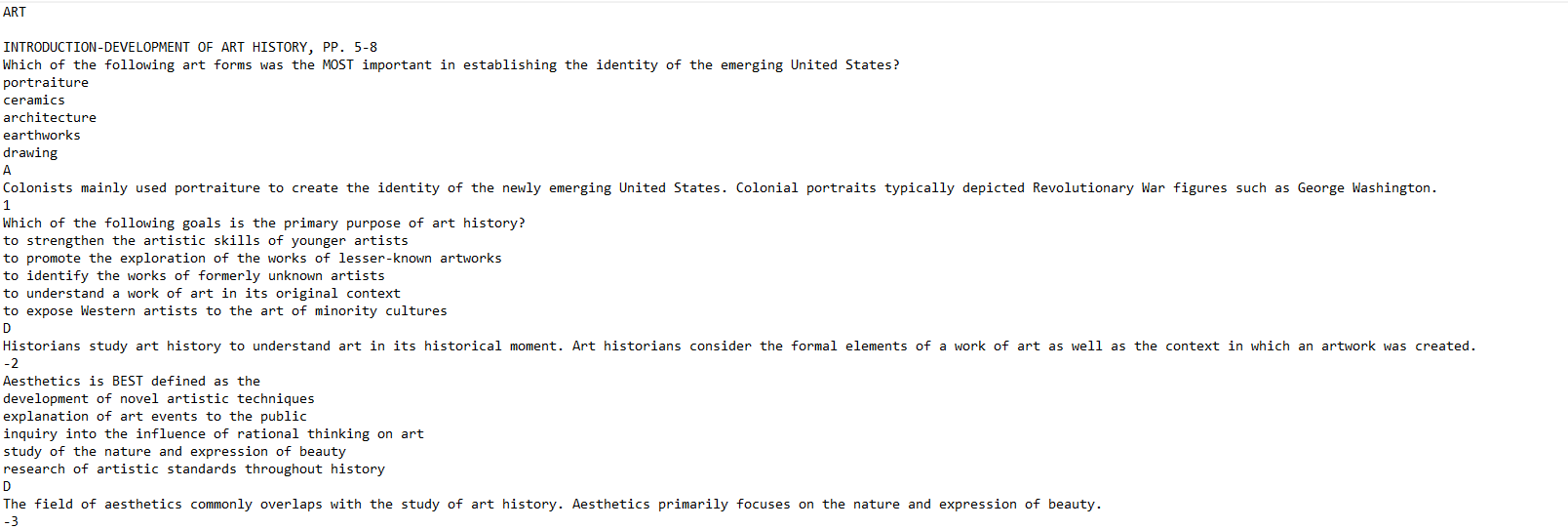
## Writing to File

### Write to Text File



PrintWriter takes all data and organizes it into a file

PrintWriter Objects are utilized in order to write to a text file and save the data. This allows the user to retain their previously imported tests and statistics upon closing the file. Furthermore, it also gives the functionality to allow the user to transfer data without importing every test again.



Test Title

Question

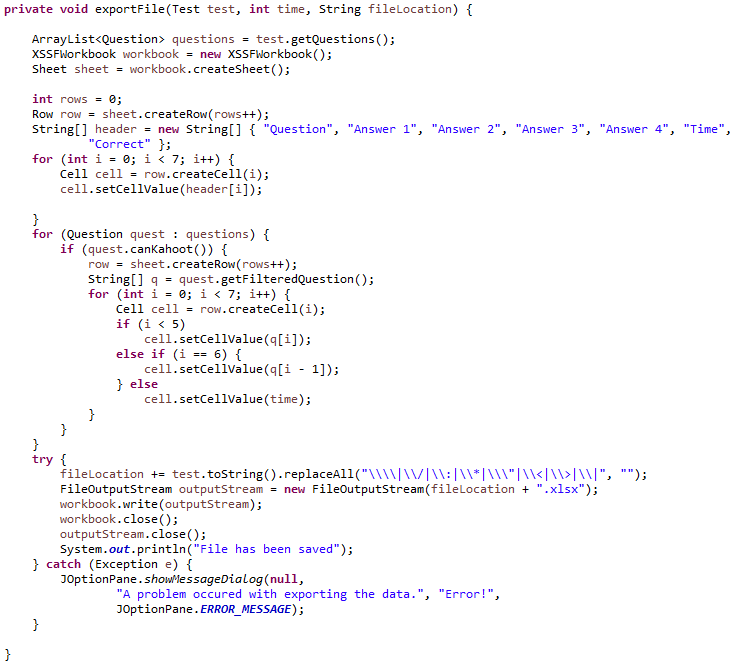
Net Score for each question

Statistics Values

Subject Dividers

A simple text file is used to store the information because it is able to hold the data in an organized and efficent way. It is split into different lines representing the specific information pertaining to each subject, test, and question.

### Write to Excel File

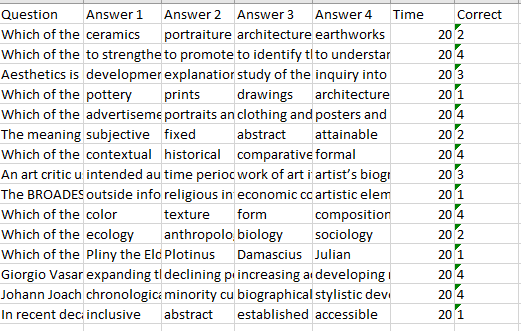


Writes to an excel file

Changes the answer choices from 5 to 4

Ensures each question fits the constraints for Kahoot

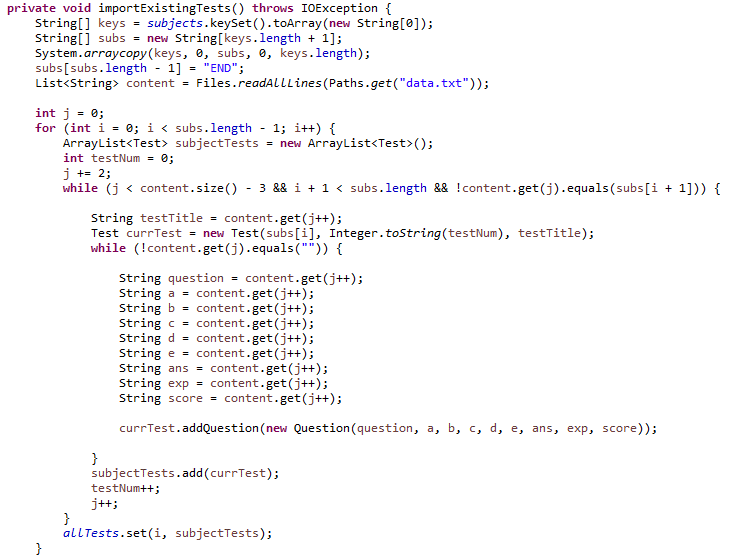
The program also utilized Apache POI’s XSSFWorkbook to create a spreadsheet with all of the data from a test and outputs it to an excel file (.xlsx). It allows the easy transfer for data into other external quizzing programs.



The file that is exported follows a specific format to be compatible with third-party programs. It separates the each question into a different row and each part of the question into a separate columns.

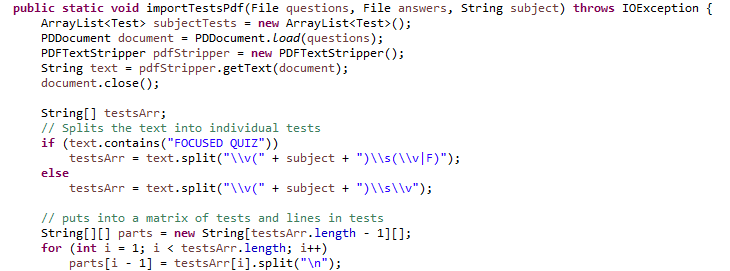
## Reading from File

### Text File



Using the Files static method to read all line, the code was easily able to separate each line of a text file into elements in a List. This allows the program to be able to read existing file when it is opened and reloads the data into the JComboBoxes and JLabels, which improves the quality of life of the program.

### PDF



Loop through all tests in the PDF

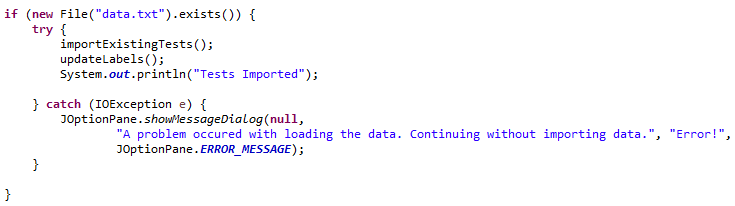


Parse one test

Importing the text from pdf files is one of the core functionalities of the program. It utilizes PDDocument and PDFTextStripper in conjunction with the String split() method in order to separate the tests and individual lines into an array. This allows for the method to easily parse the data into the format that allows for the rest of the program to function.

# Error Handling

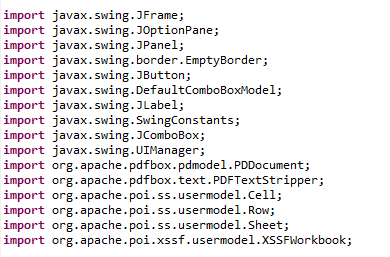
## Try Catch



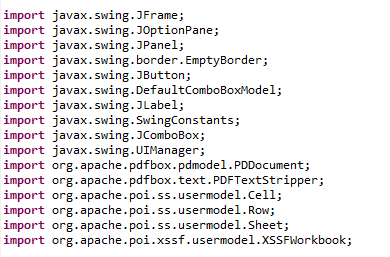
## Handling Null Values

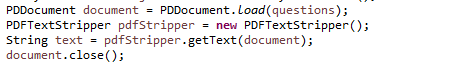
# External Libraries

## Apache Poi

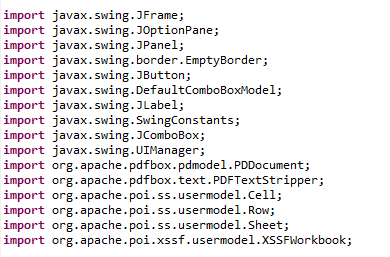


## Apache PDFBox





## Javax



# Encapsulation

# Inheritance