Andrew Rankins Bell

Katie Burgress

Professor Mohan

CMPSC 201

4 December 2021

Final Project: Progress Report

Our two person team took on the monumental task of combining two programming languages together to solve a realistic problem in our world. The task is to emulate a grocery store using Python and Java code, combined using a bash file. We have five files total, one bash and two for both Python and Java.That problem is there is a need for a tool that is able to determine what goods prices are most affordable in this ever changing world of inflation. The prices of goods and services have been on a steady incline throughout the start and length of the pandemic that has ravaged the world at large. This tool can help to support families that are most in need of resources and find prices that are sustainable. This tool could be combined with bargain hunting to positively impact lives. With a sense of urgency and determination our team began to brainstorm what would be the best approach to this enormous problem. We initially wanted to use a mixture of multiple programming languages. In this plan we decided that we wanted to use multiple languages because of how different the compilation of these languages were. We planned to use a mixture of scripting and object-oriented programming languages. We quickly found out that we already had to use a total of two different languages to create a way to connect the project together using the bash script. We were planning to implement more programming languages but then decided to only work with Java and Python.

The first programming language that was used was Python. Python is a powerful scripting language. Python was used within our tool to create a file generator that would generate an itemlist.txt file with 10 randomized products and create a purchasehistory.txt file if there wasn't one existing already.. The second language used was Java. Java is an object-oriented programming language. The java file would be where the program would be executed from. The Python program was implemented to create a randomized data file, giving random prices and ratings using the random library. The ten good names were stored inside of a list, which was iterated through in order to create the ten products. The product names and ID's were not randomized like the prices and ratings were. We began with the creation of the Python program that is the constructor for the products that would be in the virtual grocery store. This constructor would be responsible for initializing the products and then give a report of the given product and its attributes. The products have various attributes such as ID, name, price and rating. This process was straightforward with Katie effortlessly springing to action to create the program.

The next thing to do was to create the program that would read in the text file that has a list of various products. This part of the process was more complicated because I was confused about how to read in files in Python. Katie was able to navigate and connect the file that had the products and the qualities of the products entailed within. The random number generator was used to randomly group together products with the name, price, ID and rating. She was also able to create a file exception that would create a file if there was not one present within the folder. Katie has a lot of experience with utilizing python and so she was able to become a power ally in creating the program.

The first Java file created was the constructor file for the products. This was used so that the products had qualities or attributes such as ID, name, price and rating. The *this* keyword was used here to utilize the variable names to create various get methods to retrieve the attributes of the product in the main method. A constructor was used in both Java and Python to differentiate between the instance and local variables to control the scope the variables had.

Things began to become more difficult as it was time to create the second program using the object-oriented programming language Java. This program is responsible for allowing the overall tool to help shoppers with their grocery shopping by providing them with suggestions about what products that the customer would be interested in getting based on the amount of money the user defined. With this process there was a reader that would read in the text file that contained a list of previously purchased goods and products. The goods would be then put into an array which would store all of the goods which were previously bought. This is where the random number generator is used to randomly select goods the grocery store shopper would be interested in buying. Through iterating through the array list of goods, a random good was chosen. Within the main method, Katie was able to fine tune the process of how the suggestion program operated by allowing for the variable money of type double to be used as user input to suggest products based off of if they have the money to buy the item. To do this she had to import several packages into java. The results are put in an output file, purchasehistory.java, through Java. Because Java is a compiled language, it is faster than Python, which is an interpreted language. This is why the file was created in Python, which is higher-leveled than Java, giving it more ease of use. Java is faster which is why it is utilizing the files Python creates and computing all the data for the project.

We then had to create the bridge between the two programming languages so that the program could communicate between the two languages. This part of the project was the most straightforward to work through. What essentially had to be done was physically creating the connection between the two programming languages Python and Java respectively. So first the Python program was interpreted. As Python is a pure interpreted language, the interpreter stays around and is active throughout the entire time the program is executed. The source program and the input from the user are processed around the same time to produce the output. Java on the other hand is a purely compiled language. This means that the source code is translated from high level language into the equivalent machine language. In this regard, there is a two step process in the compilation. First the source code goes through a compiler to become the target program. Then the user interacts with the program providing input and is given output.

An unexpected challenge that did seem to arise was the issue of trying to use the append method in Java. The reason this was so challenging was because unlike Python with a builtin api for using the append method, Java is different. Java requires a lot of packages that need to be imported into the program by the developer. The problem was that there was a lot of usability for appending lists together. So this required some thinking outside of the box to implement. After some time and meeting Katie discovered that through importing packages such as file reader and print writer from java.io, she was able to append to a file, and was able to use ArrayLists instead of a String list in order to use the add function for ArrayLists, which behaves in the same way append does.Java and Python can be similar but are actually fundamentally different from one another. The process of calling the append method in Python was fairly simplistic in comparison to Java which requires a large amount of imported packages. At times it became a little bit of a shock to utilize the two uniquely different programming languages. Python is powerful for its ability to create scripts that are lightweight.. Which in turn makes it a higher-leveled programming language than Java on the other hand is also powerful in its own right because of how fast Java can be on a larger scale.

Another issue that showed up throughout the working process was the structure of the programs and how they would interact with each other. We had not accounted for having classes in either Java or Python so there was a need to implement a product class in both of those programming languages. By taking this approach we were able to add in easier readability. In turn this reinforced proper coding habits in object-oriented programming. The pathing of the programs and materials were organized differently with the folders. The data folder contained the text files that would be read in by the tool. The src folder housed all the programs created. The filing structure was changed because it had caused changes in the expected bridge format.

Ultimately, the programming part of the project was successful. As a team we need to now shift focus on the presentation of the tool. We have to also delve into completing the technical report to summarize the methodology of the implementation of the tool.