Michael Weeks

Jimmy Tran

OOP Programming Project Research Paper

We decided to make a program that contains our favorite practice exercises that we covered this semester. We chose this program because it allowed us to apply everything we learned throughout the different practice exercises and combine them into one single project. As mentioned, this program acts as a menu based catalog which allows the user to select from a list of different practice exercises. The program can be broken down into nine different classes, seven of which contain different practice exercises, a class that has a collection of different functions that were used throught the rest of the program, and a main driver class that acts as the menu interface to access the rest of the exercises. The different practice exercises include:

-Time Converter Exercise: This is a Java class named TimeConverter that provides methods to convert an input value of minutes into an equivalent value in hours and minutes. The program uses the JOptionPane class to prompt the user for an input value of minutes, and then converts the input value to hours and minutes by using the modulus operator to calculate the remainder of minutes and integer division to calculate the number of hours. The main() method creates an ExecutorService object and calls the PromtPaneMessage() method to prompt the user for an input value of minutes. The input value is then passed to the ConvertInputIntToHours() method to convert the value to hours and minutes, which are then displayed to the console using System.out.println(). Finally, the main() method shuts down the ExecutorService object. The PromtPaneMessage() method prompts the user to input a number of minutes greater than 60 using the JOptionPane class. If the user inputs a value less than or equal to 60, the method displays an error message and prompts the user to try again. Once the user inputs a valid value, the method returns the input value. The ConvertInputIntToHours() method takes the input value of minutes as a parameter and uses the modulus operator to calculate the remainder of minutes and integer division to calculate the number of hours. The method then displays the equivalent value in hours and minutes to the console using System.out.println().

-PrimeNumbers: This is a Java class named PrimeNumbers that finds all prime numbers from 1 up to a user-inputted value. The program uses a do-while loop to repeatedly ask the user to input a positive integer and then finds all prime numbers up to that integer using the findPrimeNumbersWithinRangeOfZeroToInputValueAndReturnInArrayList() method. The program then displays the prime numbers to the console using the processDisplayAndFormat() method. The main() method declares variables and initializes the Scanner object for user input. It then uses a do-while loop to repeatedly prompt the user to input a positive integer and find all prime numbers up to that integer using the findPrimeNumbersWithinRangeOfZeroToInputValueAndReturnInArrayList() method. The prime numbers are then displayed to the console using the processDisplayAndFormat() method. The program then prompts the user if they want to quit and if they input 'y', the loop terminates. The processDisplayAndFormat() method takes the input value and list of prime numbers as parameters and displays the list of prime numbers to the console in a formatted manner. The askUserToInputANumberAndReturnThatValue() method prompts the user to input a positive integer and returns the input value. The findPrimeNumbersWithinRangeOfZeroToInputValueAndReturnInArrayList() method takes the user input value as a parameter and finds all prime numbers from 1 up to that value. The method uses nested for loops and a label to break out of the outer loop when a non-prime number is found. The prime numbers are then stored in an ArrayList and returned. The program also provides a message to the console to indicate that it taught the user about using loops, specifically for loops to iterate through numbers and either while or do-while loops to allow the user to repeatedly use the program.

-LottoSimulator: This program simulates a lottery game, prompting the user to input three numbers within a range of 0-99. It uses the Random Class to generate a random number from 0 to one less than the entered number. It then uses loops to increment the number of draws until the winning conditions are met. The program also performs input validation to ensure the numbers entered by the user are within the range of numbers set in the program. The program outputs the number of tries it took to generate the three winning numbers.

FileIO: This program is a simple implementation of file input/output and basic cryptography. It allows the user to input a message, save it to a plain text file, encrypt it with a key provided by the user, and save the encrypted message to a new file. The program prompts the user for input at various stages, such as obtaining the key from the user and asking for a file name for the plain text file. It also includes error handling for incorrect user input. At the end, the program prints a message indicating that the files will be generated after the main program completes, and provides a summary of what was learned.

-data: This is a Java program that defines a class called "data". It contains methods for displaying a menu, executing commands based on user input, inputting numbers into an ArrayList, displaying the numbers in the ArrayList, searching for a specific number in the ArrayList, and deleting all numbers from the ArrayList. The program uses HashMaps and lambdas to execute the appropriate commands based on user input. The ArrayList of numbers is declared as a private member variable of the class.

-Main: This is a Java program that allows the user to select from a list of options, where each option corresponds to a different program (e.g. Time Converter, Lotto Simulator, Prime Numbers). The program displays a menu, takes user input, and executes the corresponding program. The user can quit the program at any time. The program uses recursion to print a message thanking a professor.

-Crypto: This program is a simple encryption and decryption tool. It has a Crypto class with methods for encrypting and decrypting a plaintext file. The encryption method takes in the name of the plaintext file, name of the output ciphertext file, and a key value. The plaintext file is read into a string, split into individual words, and each character in each word is shifted by the key value to create a new encrypted character. The encrypted characters are written to the output file. The decryption method takes in the name of the ciphertext file, name of the output plaintext file, and the same key value used in encryption. The ciphertext file is read and each character is shifted back by the key value to get the original character. The resulting plaintext words are written to the output file. The program uses ArrayList and Scanner classes to handle the data

ArrayInClass: This is a Java program that creates an instance of a FunctionLibrary class and a data class, and prompts the user to select an option from a menu using a while loop and a Scanner object. The user's input is validated using a method from the FunctionLibrary class, and if it is valid, a method from the data class is executed based on the user's input. The program also includes a message that provides an overview of the concepts covered in the exercise.

Each of the different practice exercises displays a short summary of what that particular exercise taught us after it is finished executing before it returns back to the main menu. This program taught us how to merge different individual exercises into a single exercise. It also allowed us to experiment with a few new data structures that we did not cover in class such as the Hash Map user selection menu in the invokeCommand function inside the Main class.

There are 9 different classes that make up this program. We used a recursive function to print the professors name on lines 86 in the Main class, the recursive function and body can be found on lines 95-101 in the main class. We used multiple loops throughout multiple class but specifically we used a while loop in the main class to display the recursion function and other messages associated with it on lines 79-93. In the Lottery Simulator class lines 147-153 we used a function called convertInputToInt that has a try catch block for exception handling to make sure there are no errors when converting a string to an integer. Below is a copy of our UML diagram

UML:

