Martin Rugani [Learnership - Curro Private College Cape Town - All Schools]

Planning Documentation

Contents

[A) Problem Description 2](#_Toc172633824)

[B) Research integration 2](#_Toc172633825)

[C) Viability Evaluation 4](#_Toc172633826)

[D) Economic Viability 5](#_Toc172633827)

[E) Solution Selection 6](#_Toc172633828)

[Solution 1 6](#_Toc172633829)

[Solution 2 11](#_Toc172633830)

[Chosen Solution 15](#_Toc172633831)

# Problem Description

UrbanFurn faces discrepancies when calculating their employees' overtime calculations. Errors with employee overtime pay are constantly surfacing which has led to frustration among the workers of the UrbanFurn factory. There have been numerous complaints from employees about overtime not accurately reflected in their paychecks which has increased tension in the workplace. A meeting with management and the representatives of employees was held where they decided to contract developers to create a program that will:

* Be implemented in 4 days
* Accurately Calculate employee overtime and deductions
* Display the user’s total income
* How much overtime they’ve earned
* Calculate total with overtime
* Calculate retirement Deductions
* Record the data into a text file
* Read the data from the text file

# Research integration

The program will calculate a user’s overtime and deductions, the program will also allow the user to register for a retirement plan that will be deducted from their total monthly income to do this the program will collect the following:

**Input**

1. Hours worked
2. Shift type (First, Second, Third)
3. Whether the user wants to opt into a retirement plan if they work the second or third shift

**Processing**

The program will use the user input to calculate the following:

1. Determine the user's hourly rates by checking the shift type
2. Calculate the total regular pay for the user
3. Determine if the user qualifies for overtime by checking if they’ve worked over 40 Hours
4. Calculate how many extra hours they’ve worked
5. Multiply their hourly rate by 1.5 to determine their overtime rate
6. Multiply their extra hours by their overtime rate to get their overtime pay
7. Add their overtime pay to their regular pay
8. If the user qualifies and opts into a retirement plan deduct 5% from their total
9. Calculate the user’s net pay by deducting their retirement fund from their total

**Output**

After processing the program will display:

1. The user's hours worked
2. The user's shift type
3. The user's pay rate
4. Their Regular pay
5. Their overtime pay
6. The total of their regular and total pay added together
7. Retirement deduction if any
8. Their Net pay

# Viability Evaluation

The viability of this project is measured by time to develop and the complexity of the code, the program has been specified to calculate different units of data which is related to employee income and deductions, most of these calculations are an automated process requiring at most 5 lines of code for each function to determine a returnable variable, this reduces the complexity of the code substantially making it fully feasible to complete development and testing within a day and at most 2 days, the learning curve and the onboarding process is expected to go smoothly due to the lack of user input required for calculations and outputs to be processed. The program can be made with current technology within it’s constraints the program can calculate transactions and store hours for multiple users and can be run multiple times to record different instances of user data and transactions

This program will help with the payroll issues that UrbanFurn faces by automating their overtime calculation process, the program aims to produce accurate data quickly and efficiently to reduce discrepancies and frustration within the factory.

# Economic Viability

Given the history of UrbanFurn’s precise record-keeping in other aspects of the organization, I believe this program will benefit the company. The program will help Mr Singh accurately record data by automating the calculation process which reduces the time it will take to process payments for employees. The program is also aimed at improving UrbanFurn’s technical and automated processes I believe improving these aspects of the company will help them integrate more employees into the workforce while reducing the load on the accounting team. Automating transactions will allow more employees to be added and accounted for with little to no discrepancies between them. This will also improve the reputation of management among employees which in turn will also improve Urbanfurn’s reputation as a workplace within Cape town due to word of mouth and positive feedback from employees. The estimated cost of this program is R30 000, I believe the organization will profit from investing this program by ensuring they accurately process transactions and reduce complaints and backlash from employees which will improve productivity and lessen costs

In summary, I believe this program will further automate the transaction process which will allow for quicker and more payments to be handled, accurate transaction amounts will also ensure that the company doesn’t lose more money than required and improve the reputation of the company as a whole.

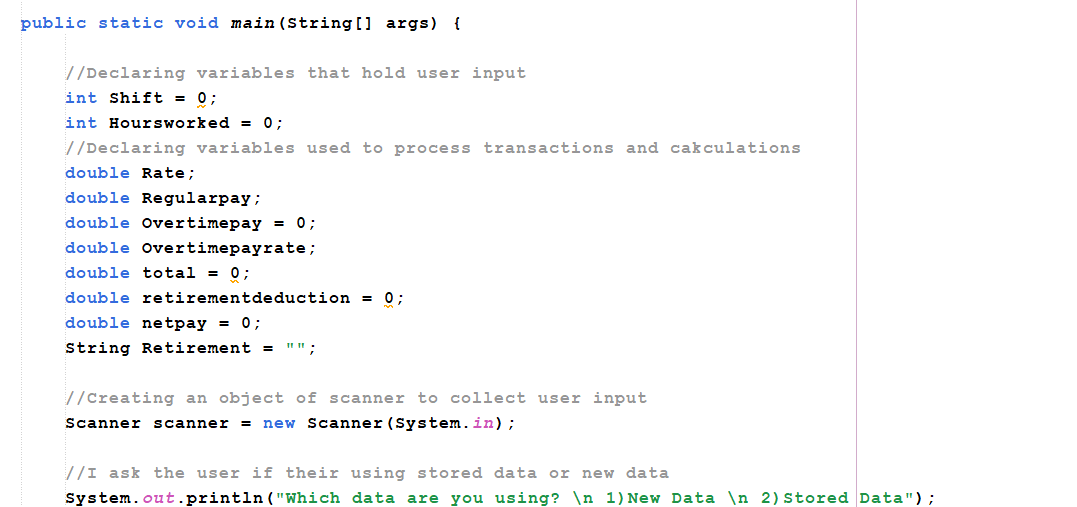
# Solution Selection

### Solution 1

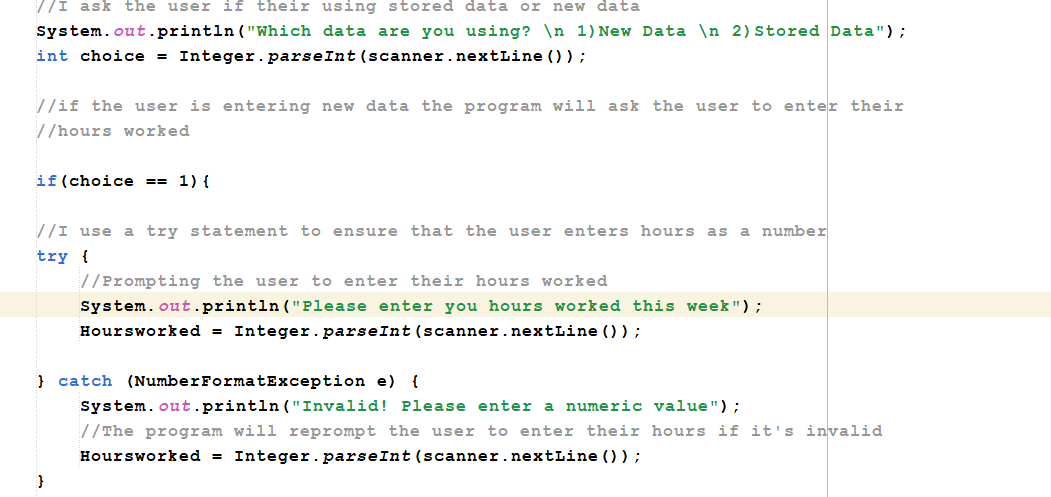
This solution aims to calculate the Overtime pay and deductions and write data to the file using randomAccessFile this method allows me to insert data anywhere in a textfile or database it also stores data as their type according to the program which can help prevent rounding errors and inaccurate translations,

This solution helps with record–keeping when data needs to be stored in a specific position in a file however this method may perform slower because it writes data types as they are in an unbuffered format.

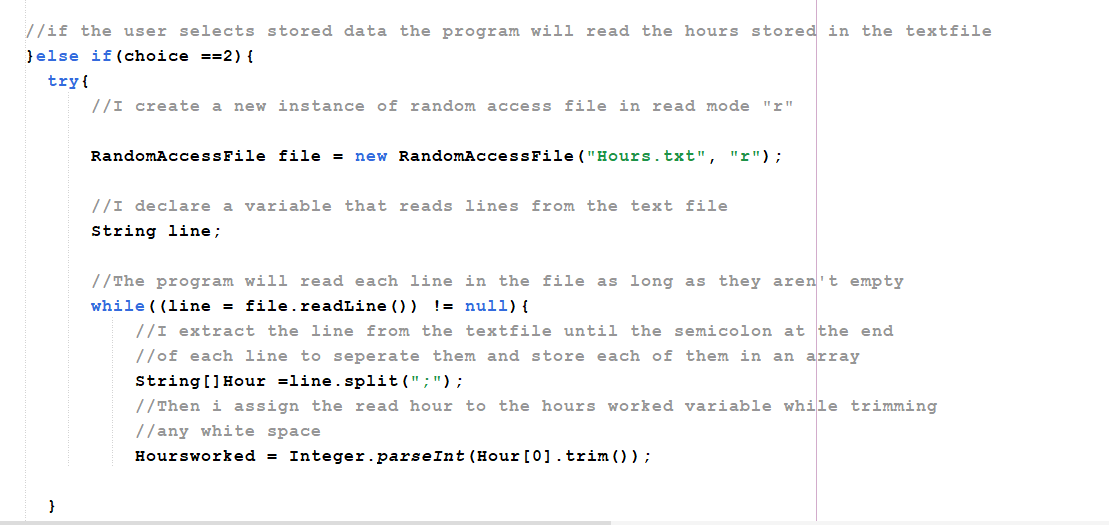
1. Storing user input



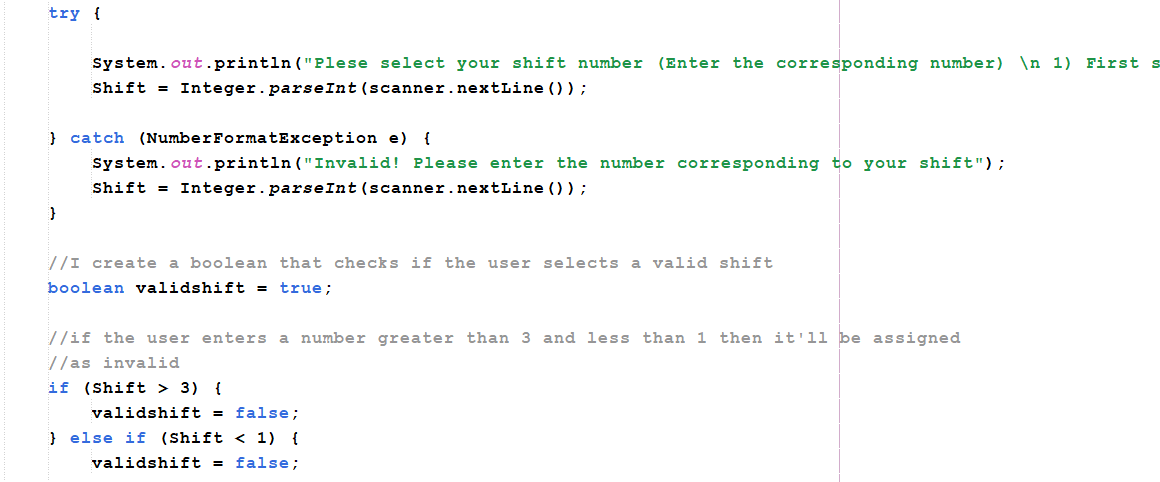
1. Collecting hours worked

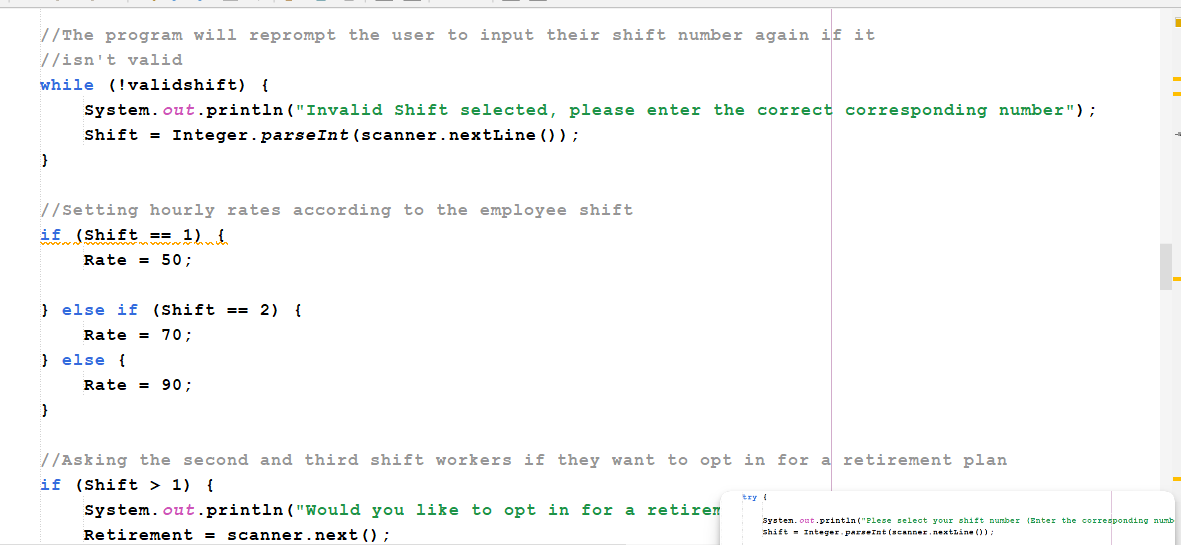


1. Reading the hours worked from the text file using RandomAccessFile

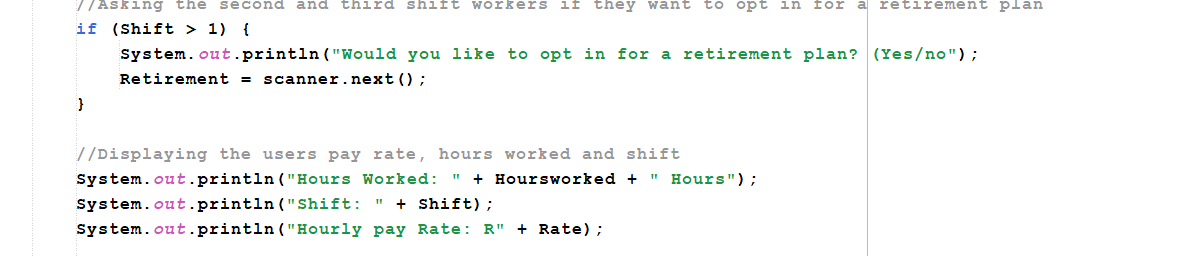


1. Validating users shift type

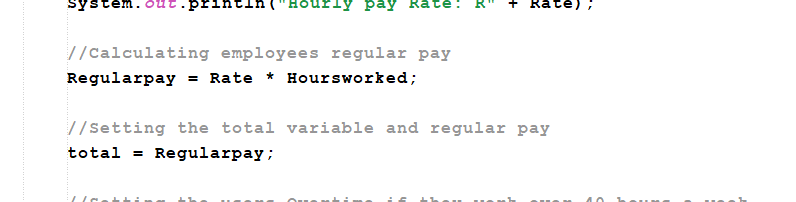




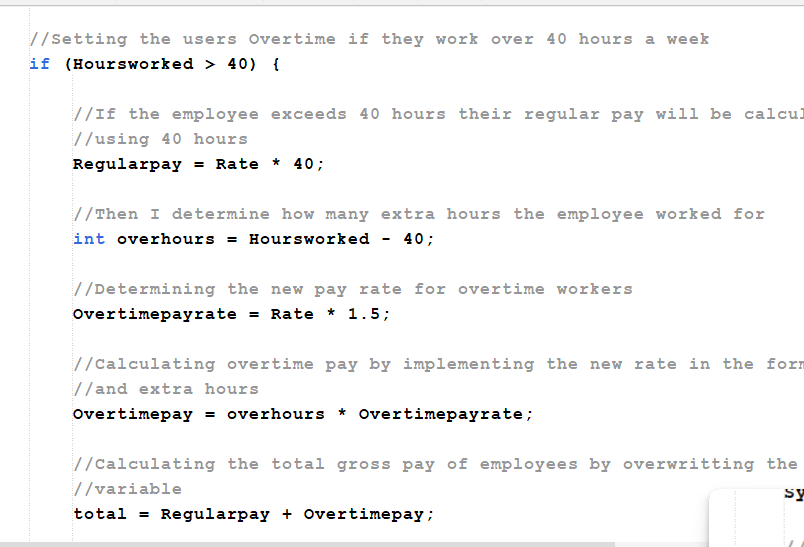
1. Asking the user if they want to opt into a retirement plan and displaying input data



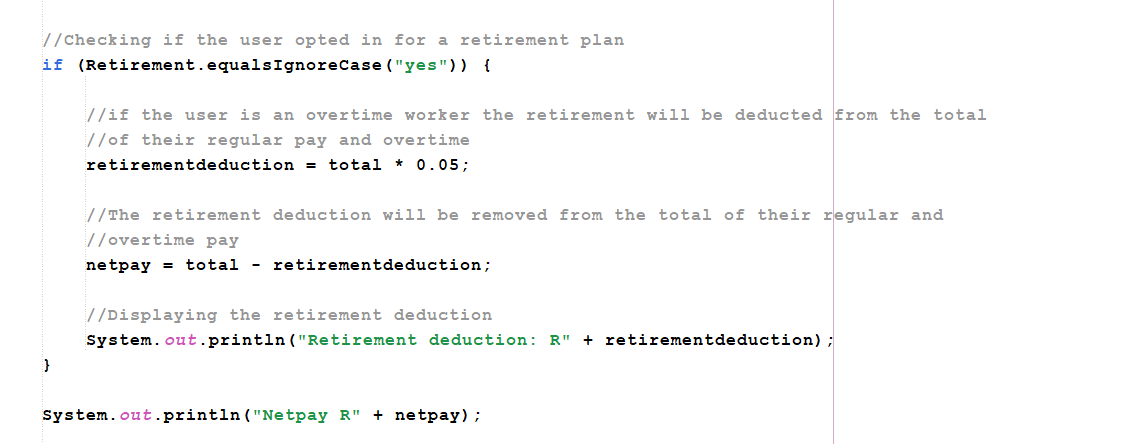
1. Calculating the users’ regular pay



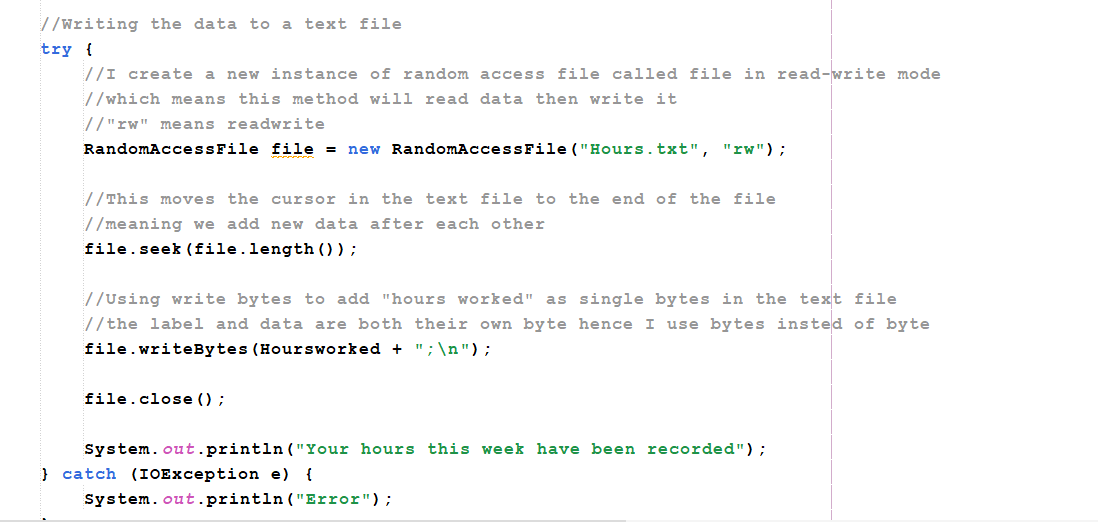
1. Calculating Overtime



1. Calculating deductions and the user’s total netpay



1. Storing the user’s hours worked into a text file using random access file

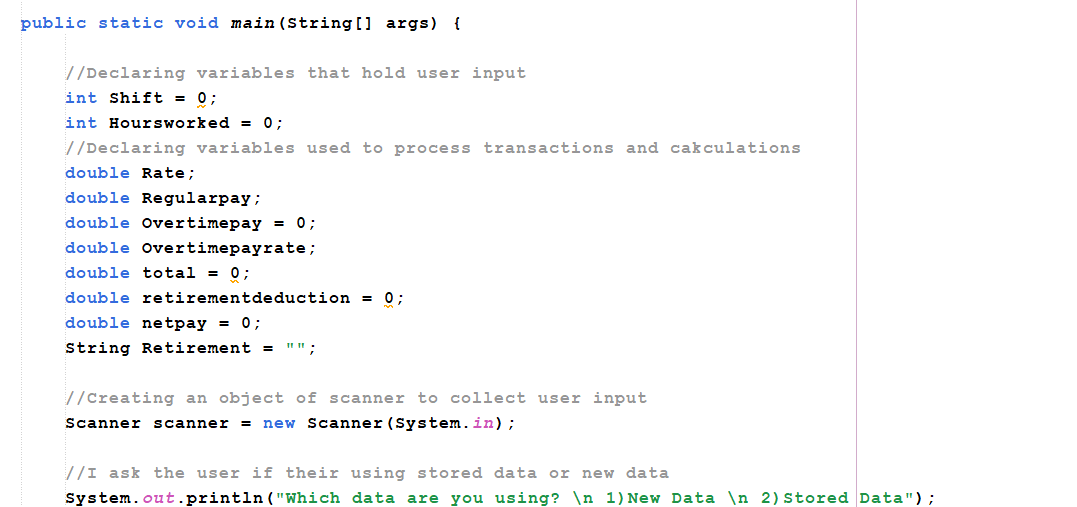


### Solution 2

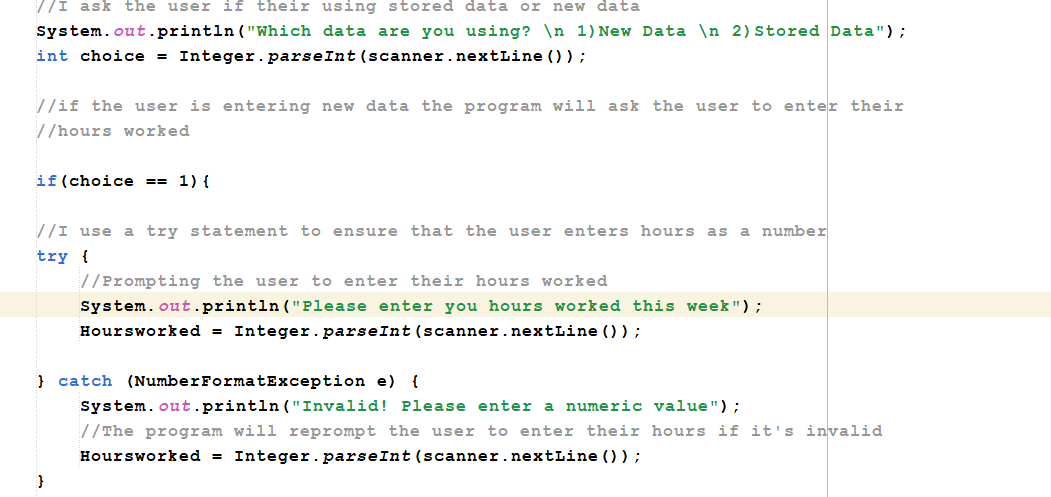
This solution will calculate the overtime pay and deductions for employees just like the previous solution, but this program differentiates itself by storing the data in an external file this process is done using filewriter and bufferedwritter these methods write data directly to a textfile to assist with record however this solution can only write data as strings or characters so it converts data in the program which could cause rounding errors or data loss.

This Method generally performs faster since it writes data as strings, but it is best used when the user wants to enter streams of characters like names or positions or when the user wants to overwrite data.

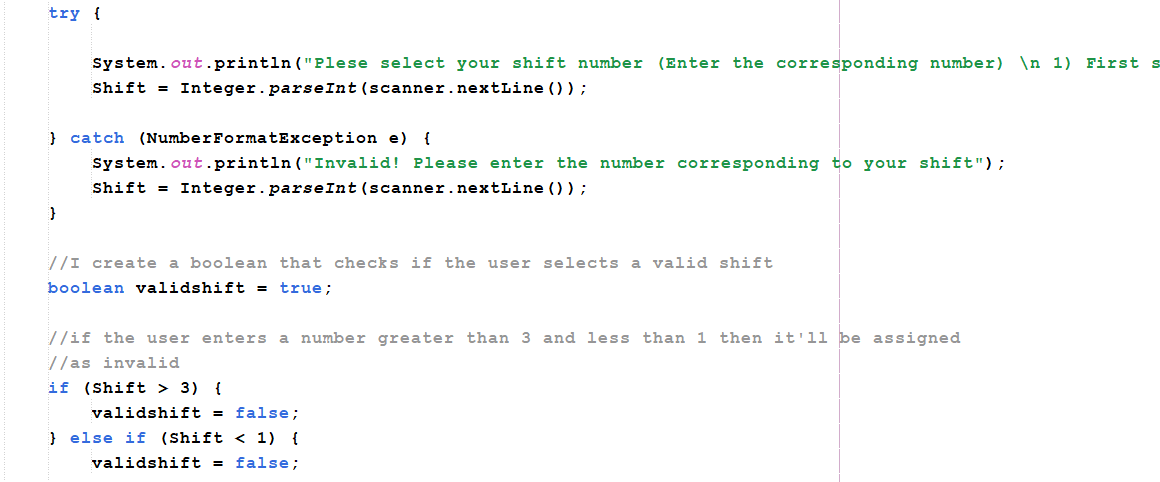
1. Storing user input

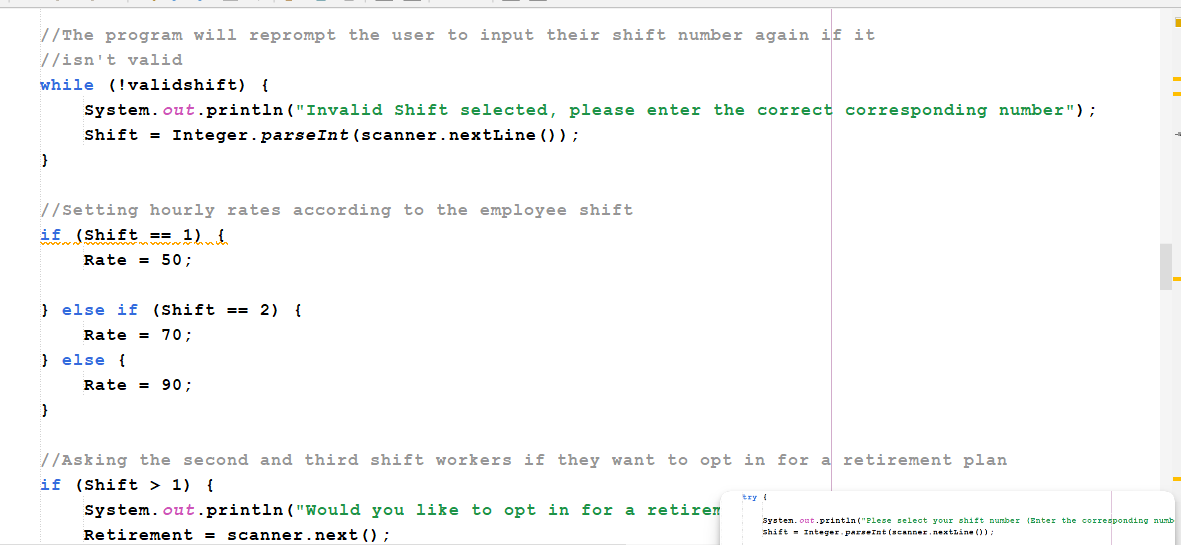


1. Collecting hours worked

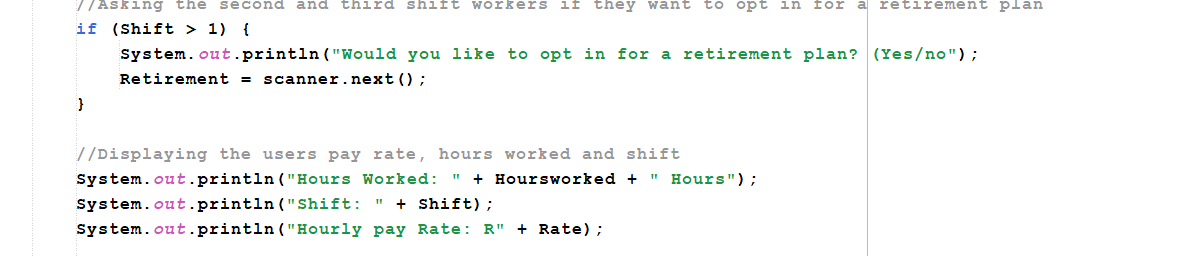


1. Validating users shift type

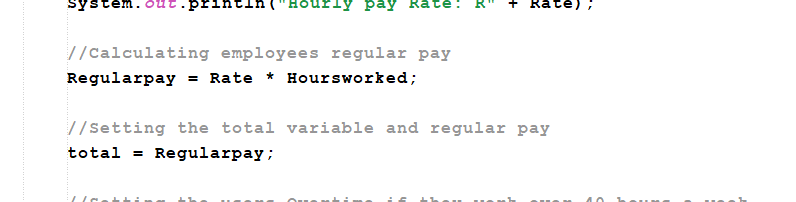




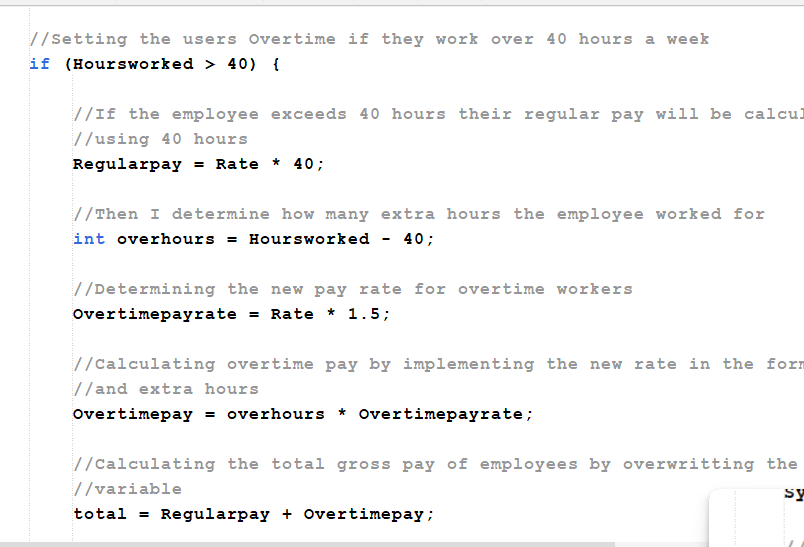
1. Asking the user if they want to opt into a retirement plan and displaying input data



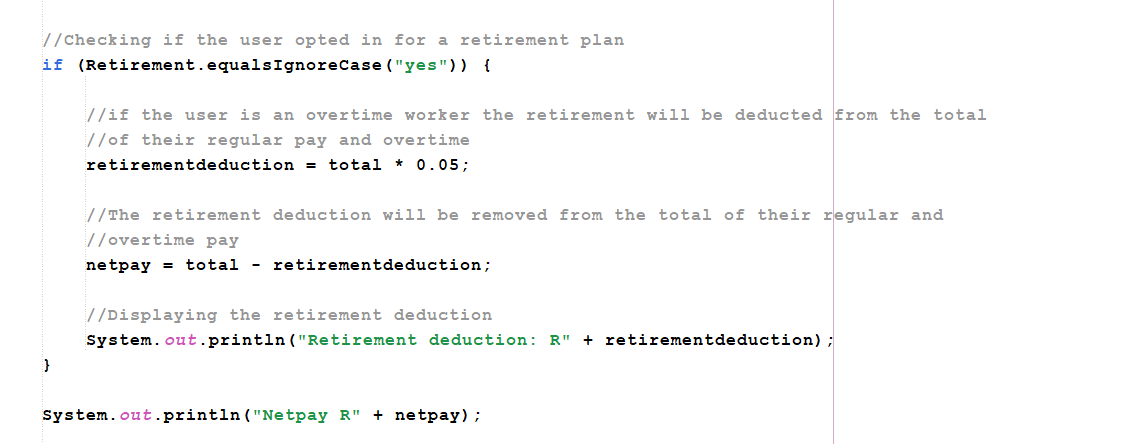
1. Calculating the users’ regular pay



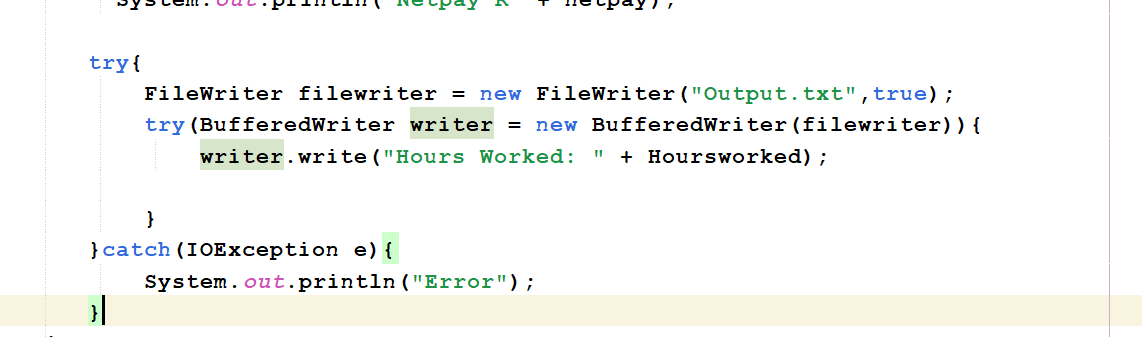
1. Calculating Overtime



1. Calculating deductions and the user’s total Net pay



1. Storing the user’s hours worked into a text file using File Writer



### Chosen Solution

In terms of complexity, the methods used in both solutions aren’t complex to implement but I chose to implement solution 1, the RandomAccessfile is a better all-rounder for adding data to a file than file writer, Random-access allows the user to enter any kind of data to a file which helps with this scenario in case the company wants to record names and transaction details.

This means that the program is scalable and can be expanded to add new data to a file when necessary, which reduces the complexity and cost of expanding the program and guarantees its longevity.

RandomAccessFile also allows data to be retrieved from any part of a file which helps when additional data that has been stored is needed for additional transactions.

The program will approximately take a week to create and implement within the system, a team of 2 developers will be assigned to develop and test the program among each other considering their salary of 12 000 a week each that brings development costs to 24 000 along with integration and device costs the total cost of this solution will be R30 000, However considering that this program automates accounting by calculating and recording data which severely lowers the workload on the accounting team which enables less costs to be assigned to them for manual bookkeeping and also allows them to focus on the funds elsewhere within the organization ensuring a return on investment