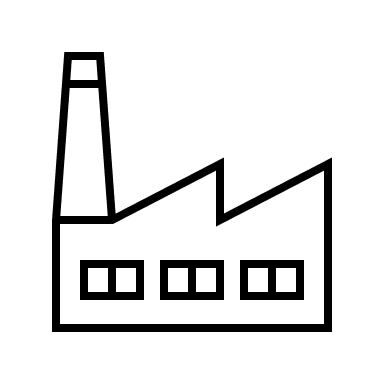
# Summative Assessment 4

# 18-23 July 2024

# Jamie Myburgh



# Documentation

# Question 6

Contents

[a) Purpose Description 3](#_Toc172627950)

[Data Flow 3](#_Toc172627951)

[b) Code Documentation 4](#_Toc172627952)

[c) READ-ME file 11](#_Toc172627953)

[d) Change Log 13](#_Toc172627954)

# Purpose Description

The program that was created is an employee payroll calculator for the company UrbanFurn, following a resurfacing error in the overtime pay department. The calculator takes user input from the employee, which is the hours that they have worked and the shift rate they were working for, and uses this to compile their net income or payroll for the week.

There are three shifts in which a user can work. The first shift pays R50 an hour, the second shift pays R70 an hour, and the third shift pays R90 an hour. The user can input any number of hours.

The standard hours worked in a week is 40 hours. Any hour that extends this is considered overtime. The overtime pay is calculated by taking the hours extending 40 and multiplying that by the rate, produced by one and one-half. This only occurs for the hours exceeding 40, not the total hours.

Should a user be working for the second or third shift, they will be offered an optional retirement plan. Should they accept the offer, 5% of their total pay will be deducted from their gross income, including their overtime pay. Should they reject the offer, the deduction will not go through.

Once the necessary information has been processed, a breakdown of the user’s payroll will be displayed. This includes their hours, their shift number, the shift rate, the regular pay, the overtime payroll if applicable, the total payroll, the retirement fund if applicable, and the net payment.

## Data Flow

The user inputs their hours. The user will then enter their shift. If the user’s hours exceed 40, the regular payroll is still calculated. The overtime payroll is also calculated, by multiplying the hours exceeding 40 by the shift rate produced by one and one-half.

Their regular income, overtime income, and net income will then be displayed/

If the second or third shift is selected, the user will be asked if they’d like to apply for the retirement fund. If the user accepts the offer, 5% of their total payroll, which includes the overtime value, is calculated. This value is then subtracted from their total payroll to acquire their net income.

If the user does not accept the retirement plan, their total payroll and net income will be displayed.

# Code Documentation

Below is the first part of the main class. Here, the necessary utilities and extensions were imported. The variables that were to be used throughout all three classes were also initialized inside the class, outside of the main method.

A close-up of a computer screen

Description automatically generated

Instances of the relevant classes were instantiated within this main method, to call upon their methods and functions.

A screenshot of a computer code

Description automatically generated

A welcome screen first displayed when the program is run. The program first prompts the user for their worked hours, and captures their response using the Scanner function, and places their value in the “hours” variable. It then prompts the user to enter their shift number, can stores their response in the “shiftInt” variable.

A screenshot of a computer

Description automatically generated

If “shiftInt” is ‘1’, then the “shift” variable is stored with the value ‘50’, if “shiftInt” is ‘2’, then the “shift” variable is stored with the value ‘70’, if “shiftInt” is ‘3’, then the “shift” variable is stored with the value ‘90’.

The calculations class was created next, extending the main class, and the necessary variables were instantiated.

A close-up of a person's face

Description automatically generated

The method to perform the calculation is opened and an if-statement is opened. If the hours exceed 40, the regular payment and overtime payment are calculated and added together and stored in the “payroll” variable.

A white background with black text

Description automatically generated

If the hours are equal to or less than 40, the shift rate and hours are multiplied together, and the result is stored in the payroll value. The method closes.

A white background with black dots

Description automatically generated

A method to display the information is created, with an if-else-statement. If the hours are over 40, it first performs the calculations and then the regular payment is displayed alongside the overtime hours. The overtime pay is also displayed with the payroll.

A screenshot of a computer program

Description automatically generated

If the hours are equal to or less than 40, it first performs the calculation and displays the payroll.

A white background with black text

Description automatically generated

The retirement class was then constructed, extending the calculations class. The necessary variables were also instantiated.

A close-up of a text

Description automatically generated

A method to perform the retirement calculations is then constructed, alongside an if-else statement. If the user’s input from the main class is equal to “yes”, it performs the calculations in the “calculations” class. It then calculates 5% of the payroll and stores it in the “retirement” variable. The method then subtracts the retirement from the payroll and stores it in the “net” variables. The deducted retirement amount and net payroll are displayed.

A text on a white background

Description automatically generated

If the user’s input is not equal to “yes”, it still performs the calculations, except it stores the “payroll” variable’s value in the “net” variable. It then prints the net-value. The method closes

A white background with black dots

Description automatically generated

This is a method to calculate the net-payment for a user who selected the first shift/is not eligible for a retirement fund. It calls the method from the “calculations” class and sets the value in the payroll-variable into the net-variable. It displays the net-value and the method closes.

A white background with black dots

Description automatically generated

Once all the classes were finished, the “main” class was adjusted accordingly. If statements were created with the necessary content. If the first shift is entered, the shift rate is set to 50. The hours, shift, and rate are displayed, and the calculations are performed by calling the method. The method to calculate the net pay is also called.

A screenshot of a computer program

Description automatically generated

If the second shift is entered, the rate is set to 70. A prompt is displayed, asking if the user would like to apply for a retirement fund. The input is collected and stored in the “input” variable. The hours, shift, and rate are printed out, and the methods to perform the calculations and retirement are called.

A screenshot of a computer

Description automatically generated

If the third shift is entered, the rate is set to 90. A prompt is displayed, asking if the user would like to apply for a retirement fund. The input is collected and stored in the “input” variable. The hours, shift, and rate are printed out, and the methods to perform the calculations and retirement are called.

A screenshot of a computer program

Description automatically generated

This is a function that stores the hours and shifts entered at each runtime. The program searches through the files for a text-filed titled “Hours and Shifts” and inputs the data here. If the file does not exist, it is created. The main method is closed, and the class is closed accordingly.

A computer screen shot of a computer program

Description automatically generated

# READ-ME file

A READ-ME.txt file has been created and is stored in the files of the program. The following displays the content within said READ-ME.txt file.

The program that was created is an employee payroll calculator for the company UrbanFurn, following a resurfacing error in the overtime pay department. The calculator takes user input from the employee, which is the hours that they have worked and the shift rate they were working for, and uses this to compile their net income or payroll for the week.

There are three shifts in which a user can work. The first shift pays R50 an hour, the second shift pays R70 an hour, and the third shift pays R90 an hour. The user can input any number of hours.

The standard hours worked in a week is 40 hours. Any hour that extends this is considered overtime. The overtime pay is calculated by taking the hours extending 40 and multiplying that by the rate, produced by one and one-half. This only occurs for the hours exceeding 40, not the total hours.

Should a user be working for the second or third shift, they will be offered an optional retirement plan. Should they accept the offer, 5% of their total pay will be deducted from their gross income, including their overtime pay. Should they reject the offer, the deduction will not go through.

Once the necessary information has been processed, a breakdown of the user’s payroll will be displayed. This includes their hours, their shift number, the shift rate, the regular pay, the overtime payroll if applicable, the total payroll, the retirement fund if applicable, and the net payment.

To run the program effectively, the following specifications are recommended, but not necessarily required:

PROCESSOR (CPU):

Intel: Core i5

MEMORY (RAM):

Recommended: 8GB

STORAGE:

Type: Solid State Drive for faster performance

Recommended Capacity: 256GB

GRAPHICS CARD:

Integrated CPU

DISPLAY:

Size: 13-14 inches (sufficient for easy usage)

Resolution: Full HD (1920 x 1080)

BATTERY LIFE:

Span: approximately 8 hours of battery life for all-day usage

OPERATING SYSTEM:

Windows

macOS

Linux

CONNECTIVITY:

Sufficient ports

Wi-Fi 6

Bluetooth 5.0

This program was created by Jamie Myburgh.

# Change Log

The if-statements for the shift-input had an additional “else” statement for invalid input. This was done so that if any other value was inputted into the system, there wouldn’t be a type-mismatch error or “build success” situation.

A close up of a text

Description automatically generated

The title of the “total” variable in the “calculations” class was changed to “regular” instead, to avoid confusion or misinterpretation of the variable’s purpose.

A close-up of a person's face

Description automatically generated

The else-if statements referring to hours less than or equal to 40 were changed to specifically mention that the hours lie between 0 and 40. This was done so that negative values were not accepted.

A white background with black dots

Description automatically generated

A white background with text

Description automatically generated

The conditions within the retirement class regarding the input were adjusted to accept only “yes” or “no”, and to otherwise display text to alert the user that the input is invalid.

A white screen with black text

Description automatically generated