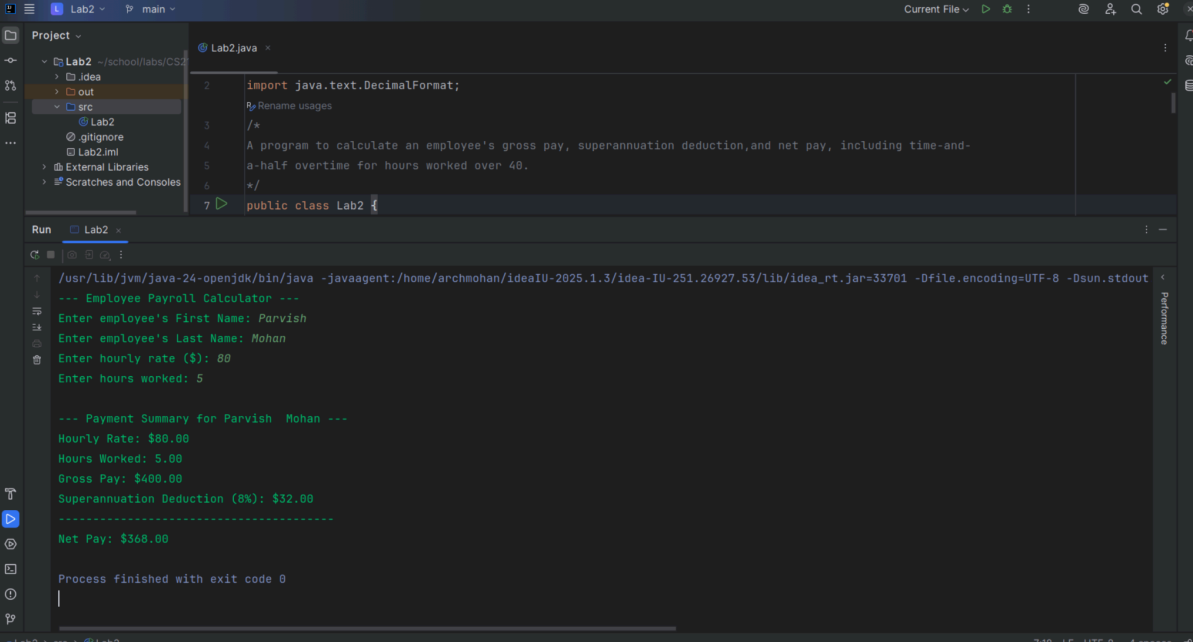


# CS218 Lab 2 - 8/8/25 Solutions

Parvish Mohan (s11230414)

## Task 1:

With an IDE



The screenshot shows an IDE window with a project named 'Lab2'. The file 'Lab2.java' is open, showing the following code:

```
2 import java.text.DecimalFormat;
3
4 /*
5  * A program to calculate an employee's gross pay, superannuation deduction, and net pay, including time-and-
6  * a-half overtime for hours worked over 40.
7  */
8 public class Lab2 {
```

The Run tab shows the following output:

```
--- Employee Payroll Calculator ---
Enter employee's First Name: Parvish
Enter employee's Last Name: Mohan
Enter hourly rate ($): 80
Enter hours worked: 5

--- Payment Summary for Parvish Mohan ---
Hourly Rate: $80.00
Hours Worked: 5.00
Gross Pay: $400.00
Superannuation Deduction (8%): $32.00
-----
Net Pay: $368.00

Process finished with exit code 0
```

Using the terminal:

```
[archmohan@x13Flow CS218]$ cd Lab
Lab1_CS218/ Lab2/
[archmohan@x13Flow CS218]$ cd Lab2
[archmohan@x13Flow Lab2]$ ls
Lab2.iml  out  src
[archmohan@x13Flow Lab2]$ cd src
[archmohan@x13Flow src]$ javac Lab2.java
[archmohan@x13Flow src]$ java Lab2
--- Employee Payroll Calculator ---
Enter employee's First Name: Parvish
Enter employee's Last Name: Mohan
Enter hourly rate ($): 40
Enter hours worked: 5

--- Payment Summary for Parvish Mohan ---
Hourly Rate: $40.00
Hours Worked: 5.00
Gross Pay: $200.00
Superannuation Deduction (8%): $16.00
-----
Net Pay: $184.00
[archmohan@x13Flow src]$
```

## Task 2: Concepts & Logic

### 1. Input analysis

Enter First and Last Name - Gets the users information

Enter hourly rate - Gets the hourly rate of the employee

Enter hours worked - Gets the hours worked by the employee

### 2. Logic Explanation

The program has a condition which checks the entered hours if it has exceeded the given threshold of 40 hours, then it calculates overtime pay when it exceeds 40, else calculates the normal gross pay.

Then, superannuation is calculated by multiplying the gross pay with the given rate of 8%.

### 3. Output Formatting

DecimalFormat - convert values into two decimal places.

Scenario 1 -

```
/usr/lib/jvm/java-24-openjdk/bin/java -javaagent:
--- Employee Payroll Calculator ---
Enter employee's First Name: Parvish
Enter employee's Last Name: Mohan
Enter hourly rate ($): 20
Enter hours worked: 38

--- Payment Summary for Parvish Mohan ---
Hourly Rate: $20.00
Hours Worked: 38.00
Gross Pay: $760.00
Superannuation Deduction (8%): $60.80
-----
Net Pay: $699.20
```

Scenario 2 -

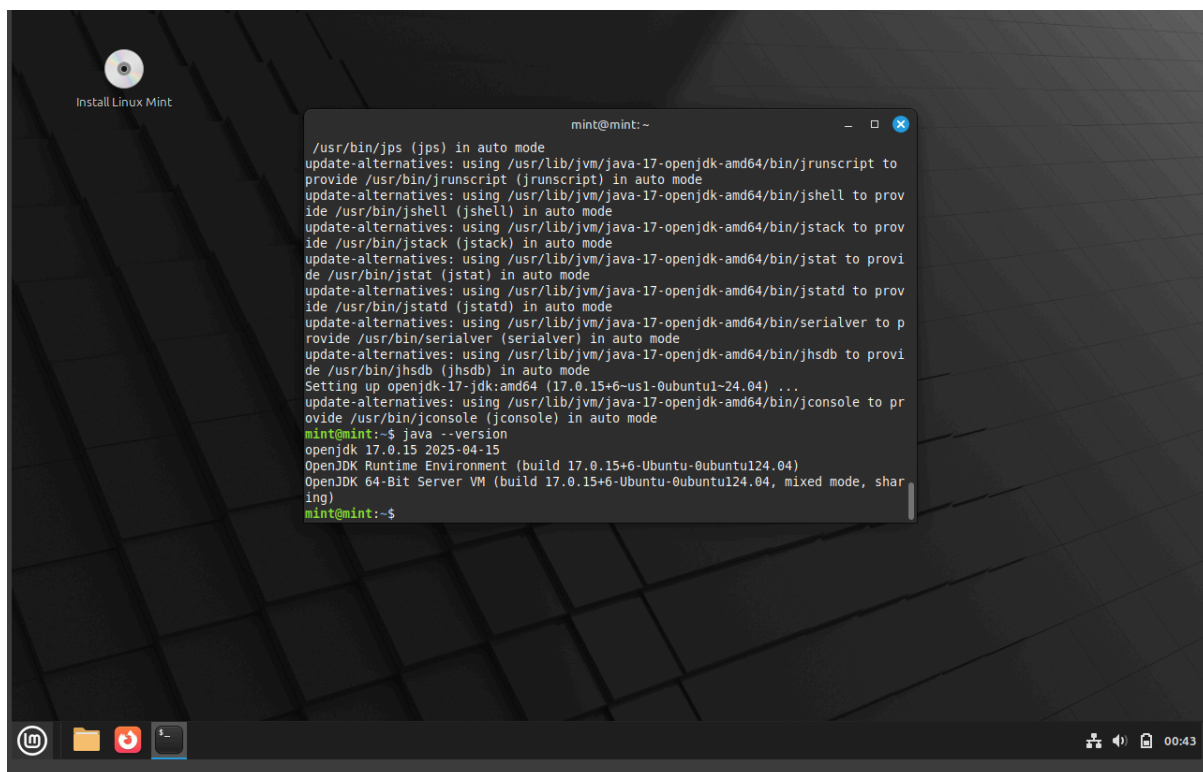
```
--- Payment Summary for Parvish Mohan ---
Hourly Rate: $20.00
Hours Worked: 42.00
Gross Pay: $840.00
Overtime Pay: $60.00
Superannuation Deduction (8%): $68.80
-----
Net Pay: $791.20
```

### Code Modification:

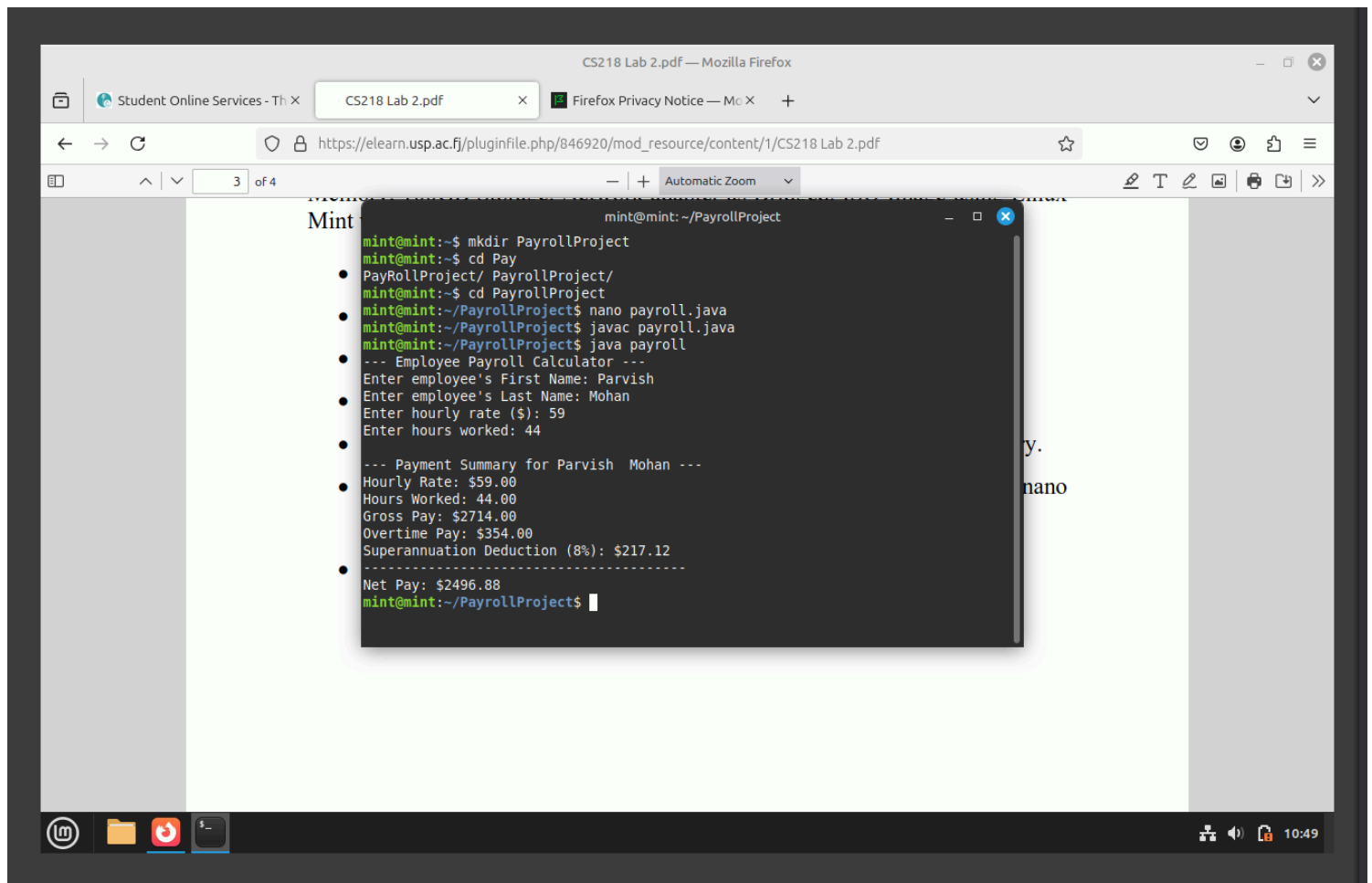
```
//tax deduction after gross pay
final double TAX_RATE = 0.05;
double tax = grossPay * TAX_RATE;
System.out.println("Tax Deduction (8%): " +
df.format(tax));
```

## Section B: Virtualization

### Running Linux Mint:



Creating a PayrollProject folder with the first exercise compiled on the linux terminal:

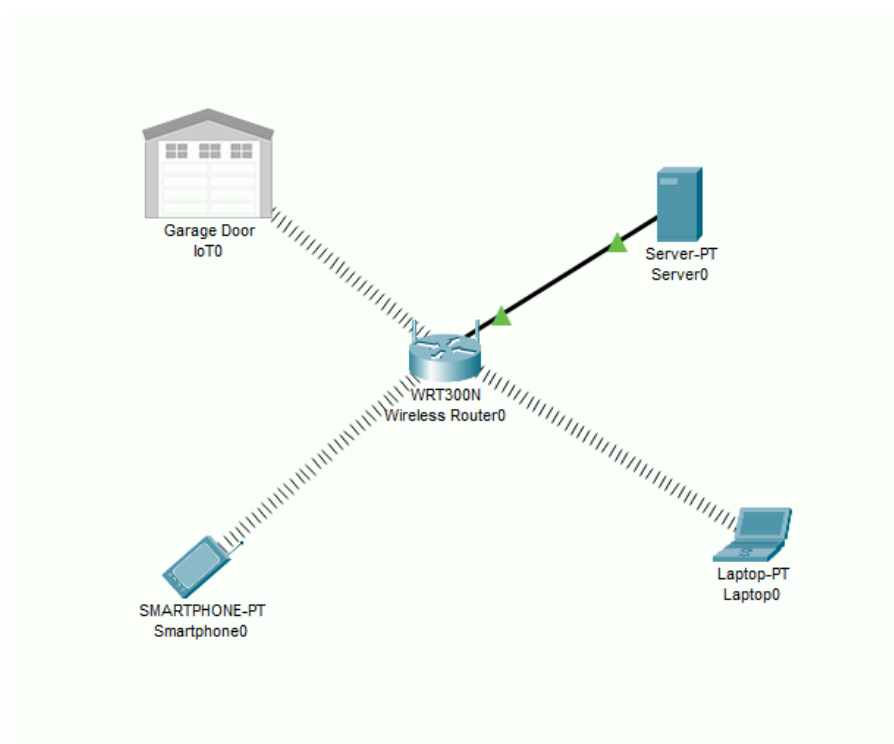


The screenshot shows a Linux terminal window with the following commands and output:

```
mint@mint:~$ mkdir PayrollProject
mint@mint:~$ cd Pay
mint@mint:~/Pay$ cd PayrollProject/
mint@mint:~/PayrollProject$ nano payroll.java
mint@mint:~/PayrollProject$ javac payroll.java
mint@mint:~/PayrollProject$ java payroll
--- Employee Payroll Calculator ---
Enter employee's First Name: Parvish
Enter employee's Last Name: Mohan
Enter hourly rate ($): 59
Enter hours worked: 44
--- Payment Summary for Parvish Mohan ---
Hourly Rate: $59.00
Hours Worked: 44.00
Gross Pay: $2714.00
Overtime Pay: $354.00
Superannuation Deduction (8%): $217.12
-----
Net Pay: $2496.88
mint@mint:~/PayrollProject$
```

## Section C: Wireless Configuration

Setup Configuration:



Final: Running the server on the smartphone using a web browser:

