COLLEGE OF BUSINESS EDUCATION



**DODOMA CAMPUS**

NATURE OF ASSIGMENT: INDIVIDUAL ASSIGNMENT

STUDENT NAME : NASRA MUSSSA NYAKUNGA

LACTURE NAME: MADAM ATUPELE KAIRO MWAITETE

REGISTRATION NUMBER : 03.5196.01.02.2023

COURSE NAME : BIT 2

SUBJECT NAME: PROGRAMING IN JAVA

**Work Hours and Payment Calculator Application**

**1. Introduction**

The "Work Hours and Payment Calculator" application is a Java-based program designed to help users track their working hours and calculate their earnings based on a specified hourly rate. The application provides a simple and user-friendly interface for logging work hours, calculating total earnings, and managing user input effectively.

**2. Features**

**2.1 Logging Work Hours**

* Users can log the number of hours they work daily.
* The program keeps a running total of all hours worked.

**2.2 Calculating Total Earnings**

* Users can calculate their total earnings based on the logged hours and their hourly rate.
* The program outputs a detailed summary, including total hours worked, the hourly rate, and total earnings.

**2.3 Menu-Driven Interface**

* The program offers a menu-driven interface with the following options:
  1. Log Hours Worked
  2. Calculate Total Earnings
  3. Exit
* The menu ensures ease of navigation and interaction.

**2.4 Exit Option**

* Users can exit the application at any time.

**3. Code Overview**

The program is implemented in Java and contains the following key components:

**3.1 Method: calculateEarnings**

* **Purpose:** Calculates total earnings by multiplying the total hours worked by the hourly rate.
* **Input:**
  + hoursWorked (int): Total hours worked.
  + hourlyRate (double): Hourly payment rate.
* **Output:**
  + Total earnings (double).

**3.2 Main Method**

* **Purpose:** Provides the main functionality of the application, including user input handling, menu display, and program control.
* **Components:**
  + A Scanner object for input handling.
  + Variables to store hourly rate, total hours worked, and total earnings.
  + A while loop to manage the menu and user choices.

**3.3 Switch Statement**

* Handles user menu choices and directs the program flow:
  + **Case 1:** Log hours worked.
  + **Case 2:** Calculate and display total earnings.
  + **Case 3:** Exit the application.
  + **Default:** Handle invalid input.

**4. Sample Run**

**Input:**

Enter your hourly rate (Tsh per hour): 5000

Menu:

1. Log Hours Worked

2. Calculate Total Earnings

3. Exit

Choose an option: 1

Enter hours worked today: 8

Hours logged successfully!

Menu:

1. Log Hours Worked

2. Calculate Total Earnings

3. Exit

Choose an option: 2

--- Earnings Summary ---

Total Hours Worked: 8 hours

Hourly Rate: Tsh 5000 per hour

Total Earnings: Tsh 40000

**Output:**

* Total Hours Worked: 8 hours
* Hourly Rate: Tsh 5000 per hour

## 5. Screenshot of the Project Interface

Below is a conceptual representation of the console interface:

Welcome to the Work Hours and Payment Calculator!

Enter your hourly rate (Tsh per hour): 5000

Menu:

1. Log Hours Worked

2. Calculate Total Earnings

3. Exit

Choose an option: 1

Enter hours worked today: 8

Hours logged successfully!

Menu:

1. Log Hours Worked

2. Calculate Total Earnings

3. Exit

Choose an option: 2

--- Earnings Summary ---

Total Hours Worked: 8 hours

Hourly Rate: Tsh 5000 per hour

Total Earnings: Tsh 40000

## 6. Challenges Faced During Development

1. **Input Validation:**
   * Ensuring the program handles invalid inputs, such as non-numeric values, was a challenge. While the program currently handles invalid menu choices, additional validation for numeric inputs could further improve reliability.
2. **User Experience:**
   * Simplifying the menu interface and maintaining a balance between functionality and user-friendliness required thoughtful design.
3. **Dynamic Input Handling:**
   * Managing continuous input through a menu-driven interface while ensuring smooth transitions between operations was a technical challenge.

## 7. Future Improvements

* Add input validation to handle invalid numeric inputs.
* Implement data persistence using file handling to store and retrieve logged hours.
* Include a graphical user interface (GUI) for enhanced user experience.
* Support additional features like tax calculations or overtime pay.

## 8. Conclusion

The "Work Hours and Payment Calculator" is a practical and straightforward Java application that addresses the need for tracking work hours and calculating earnings. By implementing a user-friendly menu-driven interface, the program ensures accessibility and ease of use for a wide audience. Further enhancements, such as input validation and file-based data storage, could make the application more robust and versatile.