Name: Sanket Banate
Roll No : 64
PRN : 202301040044
Batch: A4

Part 1(1)

Write a program by creating an 'Employee' class having the following methods and print the final salary. (Use Methods and Constructors)
1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as parameter
2 - 'AddSal()' which adds $10 to salary of the employee if it is less than $500.
3 - 'AddWork()' which adds $5 to salary of employee if the number of hours of work per day is more than 6 hours.

```java
public class Employee {
    private double salary;
    private int hoursWorkedPerDay;

    public Employee(double initialSalary, int initialHoursWorked) {
        salary = initialSalary;
        hoursWorkedPerDay = initialHoursWorked;
    }

    public void getInfo() {
        System.out.println("Salary: $" + salary);
        System.out.println("Hours worked per day: " + hoursWorkedPerDay + " hours");
    }

    public void addSal() {
        if (salary < 500) {
            salary += 10;
        }
    }

    public void addWork() {
        if (hoursWorkedPerDay > 6) {
            salary += 5;
        }
    }

    public void calculateFinalSalary() {
        addSal();
        addWork();

        System.out.println("Final Salary: $" + salary);
    }
    public static void main(String[] args) {

        Employee employee = new Employee(515, 9);

        System.out.println("Employee Information:");
        employee.getInfo();
        System.out.println("Calculating Final Salary...");
        employee.calculateFinalSalary();
    }
}
```

OUTPUT :

```
PROBLEMS 7   PORTS   OUTPUT   DEBUG CONSOLE   TERMINAL   SQL CONSOLE                    Code - Practical 2(Assignment)

cd "/Users/sanketbanate/Downloads/SY (Semester 1)/SDL JAVA/LAB/Practical 2(Assignment)/" && javac Employee.java && java Employee
sanketbanate@Sankets-MacBook-Air Java % cd "/Users/sanketbanate/Downloads/SY (Semester 1)/SDL JAVA/LAB/Practical 2(Assignment)/" && javac Employee.java && java Employee
Employee Information:
Salary: $515.0
Hours worked per day: 9 hours
Calculating Final Salary...
Final Salary: $520.0
sanketbanate@Sankets-MacBook-Air Practical 2(Assignment) %
```

PART 2 (1)

Lets create a bank account. Create a class named 'BankAccount' with the following data members
1 - Name of depositor
2 - Address of depositor
3 - Type of account
4 - Balance in account
5 - Number of transactions
Class 'BankAccount' has a method for each of the following 1 - Generate a unique account number for each depositor
For first depositor, account number will be BA1000, for second depositor it will be BA1001 and so on
2 - Display information and balance of depositor
3 - Deposit more amount in balance of any depositor
4 - Withdraw some amount from balance deposited
5 - Change address of depositor
After creating the class, do the following operations
1 - Enter the information (name, address, type of account, balance) of the depositors. Number of depositors are to be entered by user.
2 - Print the information of any depositor.
3 - Add some amount to the account of any depositor and then display final informaion of that depositor
4 - Remove some amount from the account of any depositor and then display final informaion of that depositor
5 - Change the address of any depositor and then display the final information of that depositor
6 - Randomly repeat these processes for some other bank accounts and after that print the total number of transactions.

```java
import java.util.Scanner;

class BankAccount {
    private static int accountCounter = 1000;
    public int accountNumber;
    private String name;
    private String address;
    private String accountType;
    private double balance;
    private int numTransactions;

    public BankAccount(String name, String address, String accountType, double balance) {
```

```
        this.accountNumber = accountCounter++;
        this.name = name;
        this.address = address;
        this.accountType = accountType;
        this.balance = balance;
        this.numTransactions = 0;
    }

    public void displayInfo() {
        System.out.println("Account Number: BA" + accountNumber);
        System.out.println("Name: " + name);
        System.out.println("Address: " + address);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: $" + balance);
    }

    public void deposit(double amount) {
        balance += amount;
        numTransactions++;
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            numTransactions++;
        } else {
            System.out.println("Insufficient balance to withdraw $" + amount);
        }
    }

    public void changeAddress(String newAddress) {
        address = newAddress;
    }

    public int getNumTransactions() {
        return numTransactions;
    }
}

public class BankAccountMain {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of depositors: ");
        int numDepositors = scanner.nextInt();

        BankAccount[] accounts = new BankAccount[numDepositors];

        for (int i = 0; i < numDepositors; i++) {
            scanner.nextLine();
            System.out.println("Enter information for depositor " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Address: ");
            String address = scanner.nextLine();
            System.out.print("Account Type: ");
            String accountType = scanner.nextLine();
            System.out.print("Initial Balance: $");
            double balance = scanner.nextDouble();
```

```
            accounts[i] = new BankAccount(name, address, accountType, balance);
        }

        System.out.print("Enter the account number to perform operations (e.g., BA1000): ");
        String accountNumber = scanner.next();

        BankAccount selectedAccount = null;

        for (BankAccount account : accounts) {
            if (("BA" + account.accountNumber).equals(accountNumber)) {
                selectedAccount = account;
                break;
            }
        }

        if (selectedAccount == null) {
            System.out.println("Account not found.");
            return;
        }

        System.out.println("Operations for account: BA" + selectedAccount.accountNumber);

        selectedAccount.displayInfo();

        System.out.print("Enter the amount to deposit: $");
        double depositAmount = scanner.nextDouble();
        selectedAccount.deposit(depositAmount);

        System.out.print("Enter the amount to withdraw: $");
        double withdrawAmount = scanner.nextDouble();
        selectedAccount.withdraw(withdrawAmount);

        scanner.nextLine(); // Consume the newline character
        System.out.print("Enter new address: ");
        String newAddress = scanner.nextLine();
        selectedAccount.changeAddress(newAddress);

        System.out.println("Updated Information for account: BA" +
selectedAccount.accountNumber);
        selectedAccount.displayInfo();

        int totalTransactions = 0;
        for (BankAccount account : accounts) {
            totalTransactions += account.getNumTransactions();
        }

        System.out.println("Total number of transactions for all accounts: " + totalTransactions);
    }
}
```

OUTPUT:

```
cd "/Users/sanketbanate/Desktop/coding/Java/" && javac BankAccountMain.java && java BankAccountMain
sanketbanate@Sankets-MacBook-Air Java % cd "/Users/sanketbanate/Desktop/coding/Java/" && javac BankAccountMain.java && java BankAccountMain
Enter the number of depositors: 4
Enter information for depositor 1:
Name: Sanket
Address: Pune
Account Type: Saving
Initial Balance: $36890
Enter information for depositor 2:
Name: Om
Address: Nashik
Account Type: Current
Initial Balance: $98753
Enter information for depositor 3:
Name: Satyam
Address: Nagpur
Account Type: Saving
Initial Balance: $38693
Enter information for depositor 4:
Name: Sujal
Address: Latur
Account Type: Saving
Initial Balance: $12097
Enter the account number to perform operations (e.g., BA1000): BA1000
Operations for account: BA1000
Account Number: BA1000
Name: Sanket
Address: Pune
Account Type: Saving
Balance: $36890.0
Enter the amount to deposit: $5467
Enter the amount to withdraw: $78
Enter new address: Solapur
Updated Information for account: BA1000
Account Number: BA1000
Name: Sanket
Address: Solapur
Account Type: Saving
Balance: $42279.0
Total number of transactions for all accounts: 2
sanketbanate@Sankets-MacBook-Air Java %
```

cases ⊗ 10 ⚠ 10 0 Connect Java: Ready Ln 7, Col 28 Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier