Test- 5(24.7.24)

SET 1

1. Develop a simple banking system that allows users to create accounts, deposit money, withdraw money, and check balance. Implement methods for account creation, deposit, withdrawal, and balance inquiry.

Methods:

- createAccount(String accountHolderName, double initialDeposit)
- depositMoney(String accountNumber, double amount)
- withdrawMoney(String accountNumber, double amount)
- checkBalance(String accountNumber)

program:

```
import java.util.HashMap;
import java.util.Map;
class Bank {
    private Map<String, Double> accounts;
    public Bank() {
        this.accounts = new HashMap<>();
    }
    public void createAccount(String accountHolderName, double initialDeposit) {
        String accountNumber = generateAccountNumber();
        accounts.put(accountNumber, initialDeposit);
        System.out.println("Account created successfully! Account Number: " + accountNumber);
    }
    public void depositMoney(String accountNumber, double amount) {
        if (accounts.containsKey(accountNumber)) {
            double balance = accounts.get(accountNumber);
            accounts.put(accountNumber, balance + amount);
            System.out.println("Deposit successful! New balance: " + accounts.get(accountNumber));
        } else {
            System.out.println("Account not found!");
        }
    }
    public void withdrawMoney(String accountNumber, double amount) {
        if (accounts.containsKey(accountNumber)) {
            double balance = accounts.get(accountNumber);
            if (balance >= amount) {
                accounts.put(accountNumber, balance - amount);
                System.out.println("Withdrawal successful! New balance: " + accounts.get(accountNumber));
            } else {
                System.out.println("Insufficient balance!");
            }
        } else {
            System.out.println("Account not found!");
        }
    }

    public void checkBalance(String accountNumber) {
        if (accounts.containsKey(accountNumber)) {
            System.out.println("Account balance: " + accounts.get(accountNumber));
        } else {
            System.out.println("Account not found!");
        }
    }

    private String generateAccountNumber() {
        // Simple account number generation (in a real system, use a more secure method)
        return String.valueOf(accounts.size() + 1);
    }
}

public class Main {
    public static void main(String[] args) {
        Bank bank = new Bank();
        bank.createAccount("John Doe", 1000.0);
        bank.depositMoney("1", 500.0);
        bank.withdrawMoney("1", 200.0);
        bank.checkBalance("1");
    }
}
```

Output:

```
Account created successfully: 1
Deposit successful: 1500.0
Withdrawal successful: 1300.0
Account balance: 1300.0//correct output
```

2. Create an expense tracker that allows users to add expenses, categorize them, and view a summary report. Implement methods to add expenses, categorize expenses, and generate reports.

Methods:

- addExpense(String description, double amount, String category)
- viewExpensesByCategory(String category)
- generateExpenseReport()

program:

Here is a simple expense tracker implemented in Java:

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
class ExpenseTracker {
    private Map<String, List<Expense>> expenses;

    public ExpenseTracker() {
        this.expenses = new HashMap<>();
    }

    public void addExpense(String description, double amount, String category) {
        Expense expense = new Expense(description, amount, category);
        if (expenses.containsKey(category)) {
            expenses.get(category).add(expense);
        } else {
            List<Expense> expenseList = new ArrayList<>();
            expenseList.add(expense);
            expenses.put(category, expenseList);
        }
    }

    public void viewExpensesByCategory(String category) {
        if (expenses.containsKey(category)) {
            List<Expense> expenseList = expenses.get(category);
            for (Expense expense : expenseList) {
                System.out.println("Description: " + expense.getDescription() + ", Amount: " + expense.getAmount());
            }
        } else {
            System.out.println("No expenses found in this category!");
        }
    }

    public void generateExpenseReport() {
        for (Map.Entry<String, List<Expense>> entry : expenses.entrySet()) {
            String category = entry.getKey();
            List<Expense> expenseList = entry.getValue();
            double totalAmount = 0;
            for (Expense expense : expenseList) {
                totalAmount += expense.getAmount();
            }
            System.out.println("Category: " + category + ", Total Amount: " + totalAmount);
        }
    }
}

class Expense {
    private String description;
    private double amount;
    private String category;

    public Expense(String description, double amount, String category) {
        this.description = description;
        this.amount = amount;
        this.category = category;
    }

    public String getDescription() {
        return description;
    }

    public double getAmount() {
        return amount;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        ExpenseTracker expenseTracker = new ExpenseTracker();
        expenseTracker.addExpense("Rent", 1000.0, "Housing");
        expenseTracker.addExpense("Groceries", 500.0, "Food");
        expenseTracker.addExpense("Utilities", 150.0, "Housing");
        expenseTracker.viewExpensesByCategory("Housing");
        expenseTracker.generateExpenseReport();
    }
}
```

Output:

```
Description: Rent, Amount: 1000.0
Description: Utilities, Amount: 150.0//correct output
```