1. import java.util.HashMap;

import java.util.Map;

class BankAccount {

private String accountNumber;

private String accountHolderName;

private double balance;

public BankAccount(String accountNumber, String accountHolderName, double initialDeposit) {

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

this.balance = initialDeposit;

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposited: $" + amount);

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawn: $" + amount);

} else {

System.out.println("Insufficient balance. Withdrawal failed.");

}

}

public void checkBalance() {

System.out.println("Account Balance for " + accountHolderName + ": $" + balance);

}

public String getAccountNumber() {

return accountNumber;

}

}

class Bank {

private Map<String, BankAccount> accounts;

public Bank() {

this.accounts = new HashMap<>();

}

public void createAccount(String accountHolderName, double initialDeposit) {

String accountNumber = Integer.toString(accounts.size() + 1);

BankAccount newAccount = new BankAccount(accountNumber, accountHolderName, initialDeposit);

accounts.put(accountNumber, newAccount);

System.out.println("Account created successfully. Account Number: " + accountNumber);

}

public void depositMoney(String accountNumber, double amount) {

if (accounts.containsKey(accountNumber)) {

accounts.get(accountNumber).deposit(amount);

} else {

System.out.println("Account not found. Deposit failed.");

}

}

public void withdrawMoney(String accountNumber, double amount) {

if (accounts.containsKey(accountNumber)) {

accounts.get(accountNumber).withdraw(amount);

} else {

System.out.println("Account not found. Withdrawal failed.");

}

}

public void checkBalance(String accountNumber) {

if (accounts.containsKey(accountNumber)) {

accounts.get(accountNumber).checkBalance();

} else {

System.out.println("Account not found. Balance inquiry failed.");

}

}

}

public class SimpleBankingSystem {

public static void main(String[] args) {

Bank bank = new Bank();

bank.createAccount("harsh", 1000);

bank.createAccount("krishna", 500);

bank.depositMoney("1", 200);

bank.withdrawMoney("1", 100);

bank.checkBalance("1");

bank.depositMoney("2", 100);

bank.withdrawMoney("2", 600);

bank.checkBalance("2");

bank.checkBalance("3");

2. import java.util.\*;

public class ExpenseTracker {

private List<Expense> expenses = new ArrayList<>();

public void addExpense(String description, double amount, String category) {

expenses.add(new Expense(description, amount, category));

}

public List<Expense> viewExpensesByCategory(String category) {

List<Expense> expensesByCategory = new ArrayList<>();

for (Expense expense : expenses) {

if (expense.getCategory().equals(category)) {

expensesByCategory.add(expense);

}

}

return expensesByCategory;

}

public void generateExpenseReport() {

for (Expense expense : expenses) {

System.out.println(expense.getDescription() + " - $" + expense.getAmount());

}

}

private static 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 String getCategory() {

return category;

}

}

public static void main(String[] args) {

ExpenseTracker tracker = new ExpenseTracker();

tracker.addExpense("Groceries", 80.0, "Food");

tracker.addExpense("Internet Bill", 50.0, "Utilities");

tracker.addExpense("Dinner", 20.0, "Food");

System.out.println("Expenses by Food category:");

List<Expense> foodExpenses = tracker.viewExpensesByCategory("Food");

for (Expense expense : foodExpenses) {

System.out.println(expense.getDescription() + " - $" + expense.getAmount());

}

System.out.println("\nExpense Report:");

tracker.generateExpenseReport();

}

}