CSA0961 – JAVA CLASS TEST 5

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 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 String getAccountNumber() {
        return accountNumber;
    }
    public String getAccountHolderName() {
        return accountHolderName;
    }
```

```
    public double getBalance() {
        return balance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited " + amount + " successfully. New balance: " + balance);
    }
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn " + amount + " successfully. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance. Withdrawal failed.");
        }
    }
}
class Bank {
    private Map<String, BankAccount> accounts;
    public Bank() {
        accounts = new HashMap<>();
    }
    public void createAccount(String accountNumber, String accountHolderName, double
initialDeposit) {
        if (accounts.containsKey(accountNumber)) {
            System.out.println("Account creation failed. Account number already exists.");
        } else {
            BankAccount newAccount = new BankAccount(accountNumber, accountHolderName,
initialDeposit);
            accounts.put(accountNumber, newAccount);
            System.out.println("Account created successfully.");
        }
    }
    public void depositMoney(String accountNumber, double amount) {
```

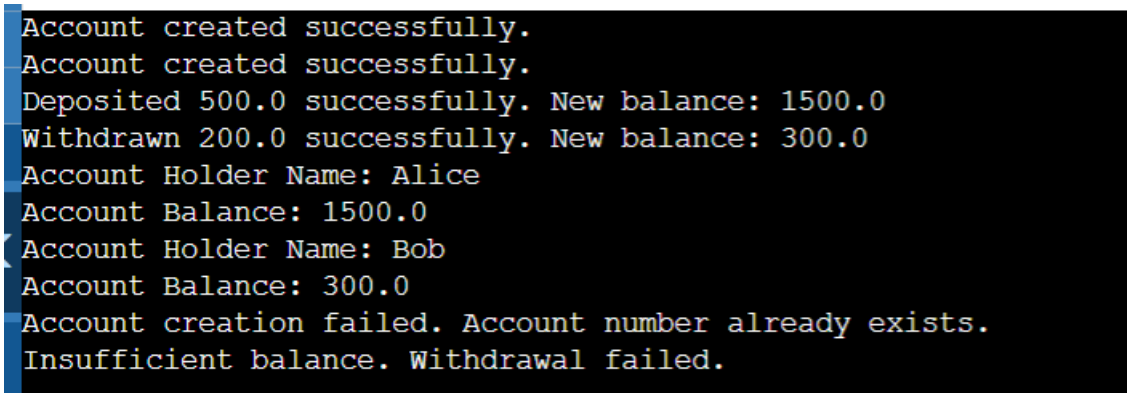
```
        if (accounts.containsKey(accountNumber)) {
            BankAccount account = accounts.get(accountNumber);
            account.deposit(amount);
        } else {
            System.out.println("Account not found. Deposit failed.");
        }
    }
    public void withdrawMoney(String accountNumber, double amount) {
        if (accounts.containsKey(accountNumber)) {
            BankAccount account = accounts.get(accountNumber);
            account.withdraw(amount);
        } else {
            System.out.println("Account not found. Withdrawal failed.");
        }
    }
    public void checkBalance(String accountNumber) {
        if (accounts.containsKey(accountNumber)) {
            BankAccount account = accounts.get(accountNumber);
            System.out.println("Account Holder Name: " + account.getAccountHolderName());
            System.out.println("Account Balance: " + account.getBalance());
        } else {
            System.out.println("Account not found.");
        }
    }
}

public class Main {

    public static void main(String[] args) {
        Bank bank = new Bank();
        bank.createAccount("123456", "Alice", 1000);
        bank.createAccount("789012", "Bob", 500);
```

```
        bank.depositMoney("123456", 500);
        bank.withdrawMoney("789012", 200);
       bank.checkBalance("123456");
        bank.checkBalance("789012");
        bank.createAccount("123456", "Charlie", 2000);
        bank.withdrawMoney("789012", 1000);
    }
}
```

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 :

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

public class ExpenseTracker {
```

```
private List<Expense> expenses;
private Map<String, List<Expense>> expensesByCategory;
public ExpenseTracker() {
    expenses = new ArrayList<>();
    expensesByCategory = new HashMap<>();
}
public void addExpense(String description, double amount, String category) {
    Expense newExpense = new Expense(description, amount, category);
    expenses.add(newExpense);
    if (!expensesByCategory.containsKey(category)) {
        expensesByCategory.put(category, new ArrayList<>());
    }
    expensesByCategory.get(category).add(newExpense);
}
public void viewExpensesByCategory(String category) {
    if (expensesByCategory.containsKey(category)) {
        List<Expense> expensesInCategory = expensesByCategory.get(category);
        System.out.println("Expenses in category '" + category + "':");
        for (Expense expense : expensesInCategory) {
            System.out.println(expense);
        }
    } else {
        System.out.println("No expenses found in category '" + category + "'.");
    }
}
public void generateExpenseReport() {
    System.out.println("Expense Report:");
    for (String category : expensesByCategory.keySet()) {
        System.out.println("Category: " + category);
        double totalCategoryExpense = 0.0;
        List<Expense> expensesInCategory = expensesByCategory.get(category);
        for (Expense expense : expensesInCategory) {
```

```
            totalCategoryExpense += expense.getAmount();
        }
        System.out.println("Total Expenses in " + category + ": $" + totalCategoryExpense);
    }
}
private 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;
    }
    @Override
    public String toString() {
        return "Description: " + description + ", Amount: $" + amount;
    }
}
public static void main(String[] args) {
    ExpenseTracker tracker = new ExpenseTracker();
    tracker.addExpense("Lunch", 15.50, "Food");
    tracker.addExpense("Movie ticket", 10.00, "Entertainment");
```

```
        tracker.addExpense("Gasoline", 40.00, "Travel");
        tracker.addExpense("Groceries", 60.00, "Food");
        tracker.viewExpensesByCategory("Food");
        tracker.generateExpenseReport();
    }
}
```

OUTPUT :

```
java -cp /tmp/nF9ewexVCe/ExpenseTracker
Expenses in category 'Food':
Description: Lunch, Amount: $15.5
Description: Groceries, Amount: $60.0
Expense Report:
Category: Travel
Total Expenses in Travel: $40.0
Category: Entertainment
Total Expenses in Entertainment: $10.0
Category: Food
Total Expenses in Food: $75.5

=== Code Execution Successful ===
```