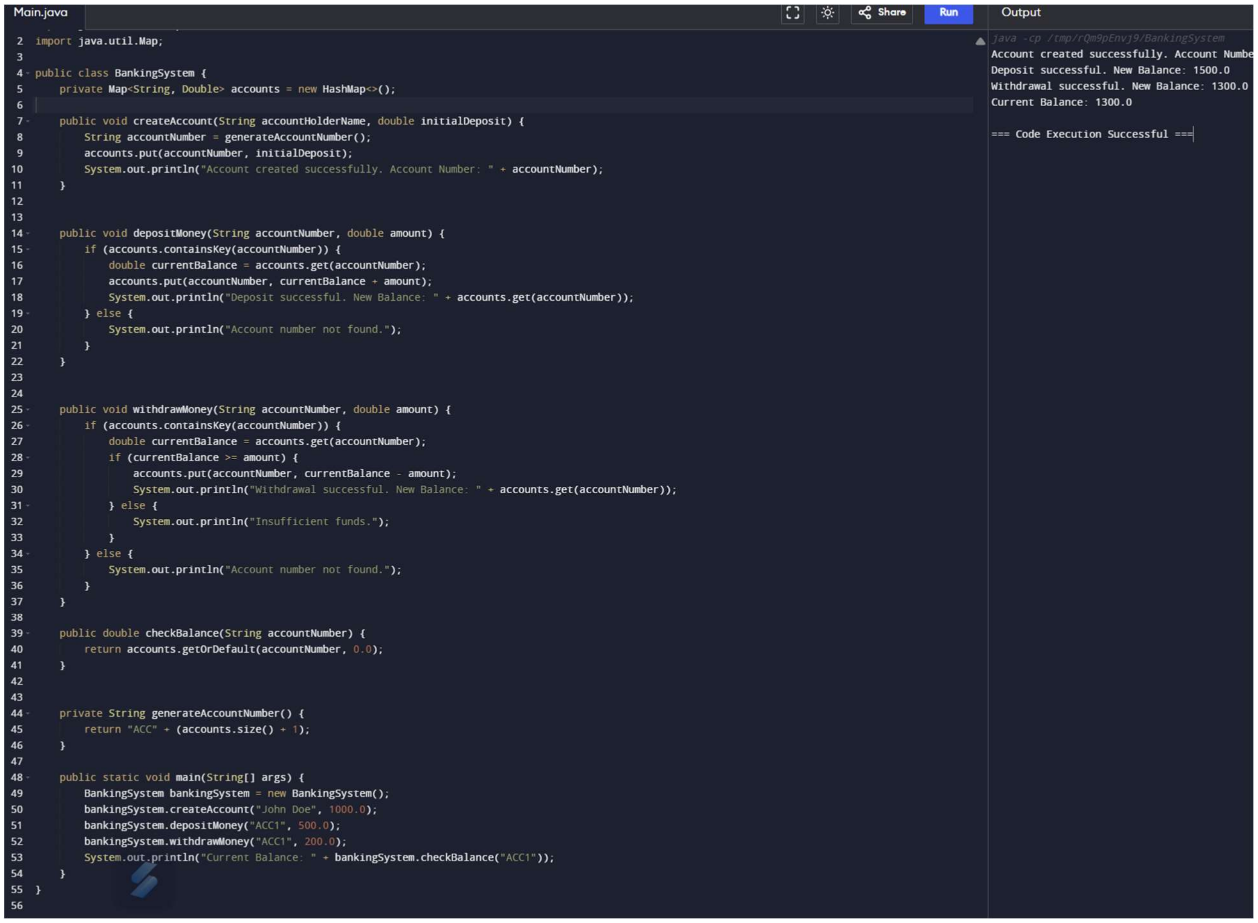
Test Question 5(24.7.24)

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)



```java
import java.util.Map;

public class BankingSystem {
    private Map<String, Double> 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 currentBalance = accounts.get(accountNumber);
            accounts.put(accountNumber, currentBalance + amount);
            System.out.println("Deposit successful. New Balance: " + accounts.get(accountNumber));
        } else {
            System.out.println("Account number not found.");
        }
    }


    public void withdrawMoney(String accountNumber, double amount) {
        if (accounts.containsKey(accountNumber)) {
            double currentBalance = accounts.get(accountNumber);
            if (currentBalance >= amount) {
                accounts.put(accountNumber, currentBalance - amount);
                System.out.println("Withdrawal successful. New Balance: " + accounts.get(accountNumber));
            } else {
                System.out.println("Insufficient funds.");
            }
        } else {
            System.out.println("Account number not found.");
        }
    }

    public double checkBalance(String accountNumber) {
        return accounts.getOrDefault(accountNumber, 0.0);
    }


    private String generateAccountNumber() {
        return "ACC" + (accounts.size() + 1);
    }

    public static void main(String[] args) {
        BankingSystem bankingSystem = new BankingSystem();
        bankingSystem.createAccount("John Doe", 1000.0);
        bankingSystem.depositMoney("ACC1", 500.0);
        bankingSystem.withdrawMoney("ACC1", 200.0);
        System.out.println("Current Balance: " + bankingSystem.checkBalance("ACC1"));
    }
}

```

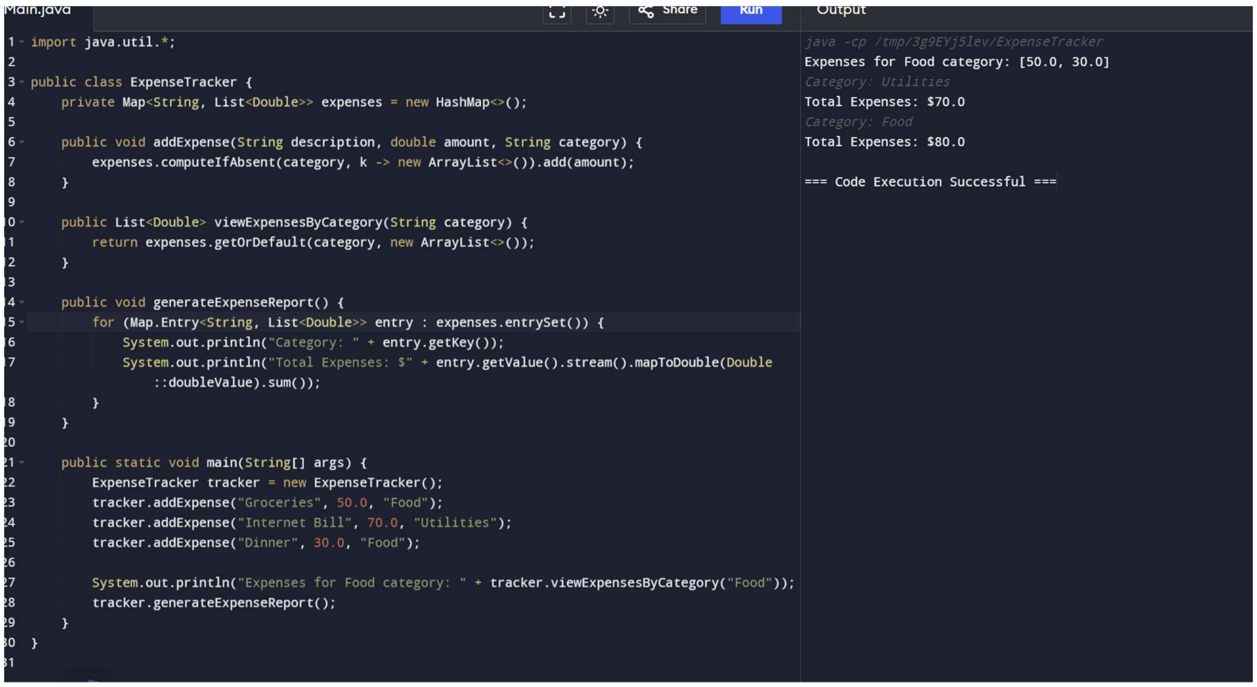
```
java -cp /tmp/rQm9pEnvj9/BankingSystem
Account created successfully. Account Numbe
Deposit successful. New Balance: 1500.0
Withdrawal successful. New Balance: 1300.0
Current Balance: 1300.0

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

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()



```java
import java.util.*;

public class ExpenseTracker {
    private Map<String, List<Double>> expenses = new HashMap<>();

    public void addExpense(String description, double amount, String category) {
        expenses.computeIfAbsent(category, k -> new ArrayList<>()).add(amount);
    }

    public List<Double> viewExpensesByCategory(String category) {
        return expenses.getOrDefault(category, new ArrayList<>());
    }

    public void generateExpenseReport() {
        for (Map.Entry<String, List<Double>> entry : expenses.entrySet()) {
            System.out.println("Category: " + entry.getKey());
            System.out.println("Total Expenses: $" + entry.getValue().stream().mapToDouble(Double
                ::doubleValue).sum());
        }
    }

    public static void main(String[] args) {
        ExpenseTracker tracker = new ExpenseTracker();
        tracker.addExpense("Groceries", 50.0, "Food");
        tracker.addExpense("Internet Bill", 70.0, "Utilities");
        tracker.addExpense("Dinner", 30.0, "Food");

        System.out.println("Expenses for Food category: " + tracker.viewExpensesByCategory("Food"));
        tracker.generateExpenseReport();
    }
}
```

```
java -cp /tmp/3g9EYj5lev/ExpenseTracker
Expenses for Food category: [50.0, 30.0]
Category: Utilities
Total Expenses: $70.0
Category: Food
Total Expenses: $80.0

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