# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

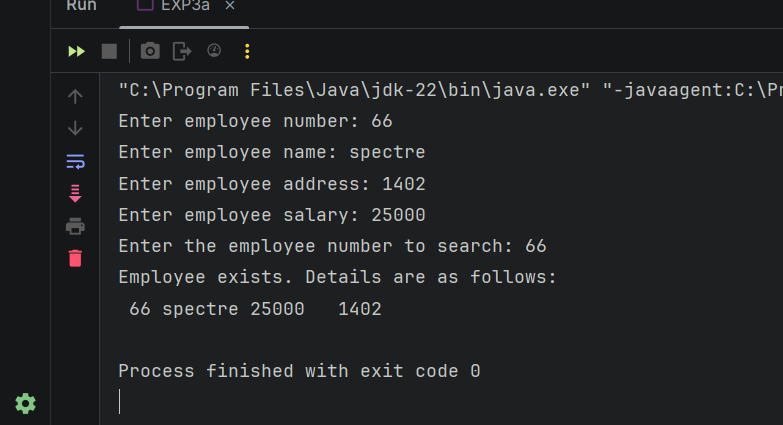
Practical No: 3

1) **Implementation of Employee class in detail operation like getEmployee(), showEmployee(), searchEmployee(), updateEmployee(), etc.**

CODE:

package EXP3;  
  
import java.util.Scanner;  
  
class emp {  
 int empNumber, empSalary;  
 String empName, empAddress;  
 static Scanner sc = new Scanner(System.in);  
  
 void getEmployee() {  
 System.out.print("Enter employee number: ");  
 this.empNumber = sc.nextInt();  
 System.out.print("Enter employee name: ");  
 this.empName = sc.next();  
 System.out.print("Enter employee address: ");  
 this.empAddress = sc.next();  
 System.out.print("Enter employee salary: ");  
 this.empSalary = sc.nextInt();  
 }  
  
 void getEmployee(int empNumber, int empSalary, String empName, String empAddress) {  
 this.empNumber = empNumber;  
 this.empName = empName;  
 this.empAddress = empAddress;  
 this.empSalary = empSalary;  
 }  
  
 void showEmployee() {  
 System.out.println(" " + empNumber + "\t" + empName + "\t" + empSalary + "\t" + empAddress);  
 }  
  
 static void searchEmployee(emp[] arr, int searchId) {  
 boolean found = false;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i].empNumber == searchId) {  
 System.out.print("Employee exists. Details are as follows: \n");  
 arr[i].showEmployee();  
 found = true;  
 }  
 }  
 if (!found) {  
 System.out.println("No such Employee exists.");  
 }  
 }  
  
 void updateEmployee() {  
 System.out.print("Please enter updated employee address: ");  
 this.empAddress = sc.next();  
 System.out.print("Enter the updated salary: ");  
 this.empSalary = sc.nextInt();  
 this.showEmployee();  
 }  
}  
  
class EXP3a {  
 public static void main(String[] args) {  
 emp[] arr = new emp[1];  
 for (int i = 0; i < arr.length; i++) {  
 arr[i] = new emp();  
 arr[i].getEmployee();  
 }  
 System.out.print("Enter the employee number to search: ");  
 int s = emp.sc.nextInt();  
 emp.searchEmployee(arr, s);  
 }  
}

OUTPUT:

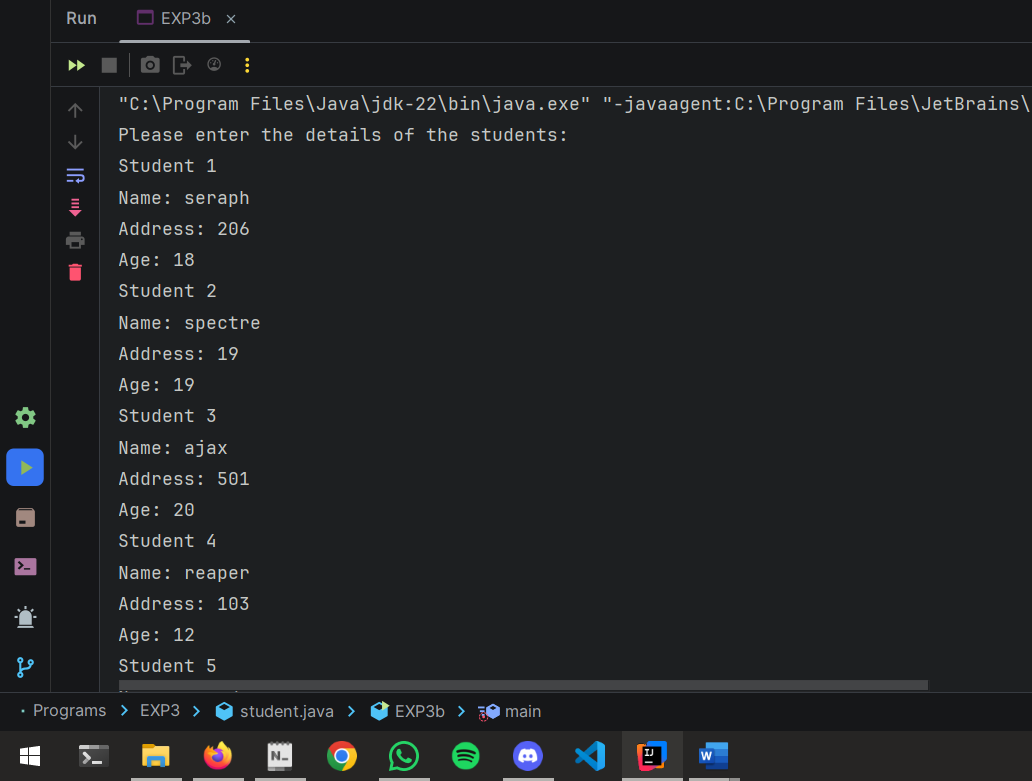


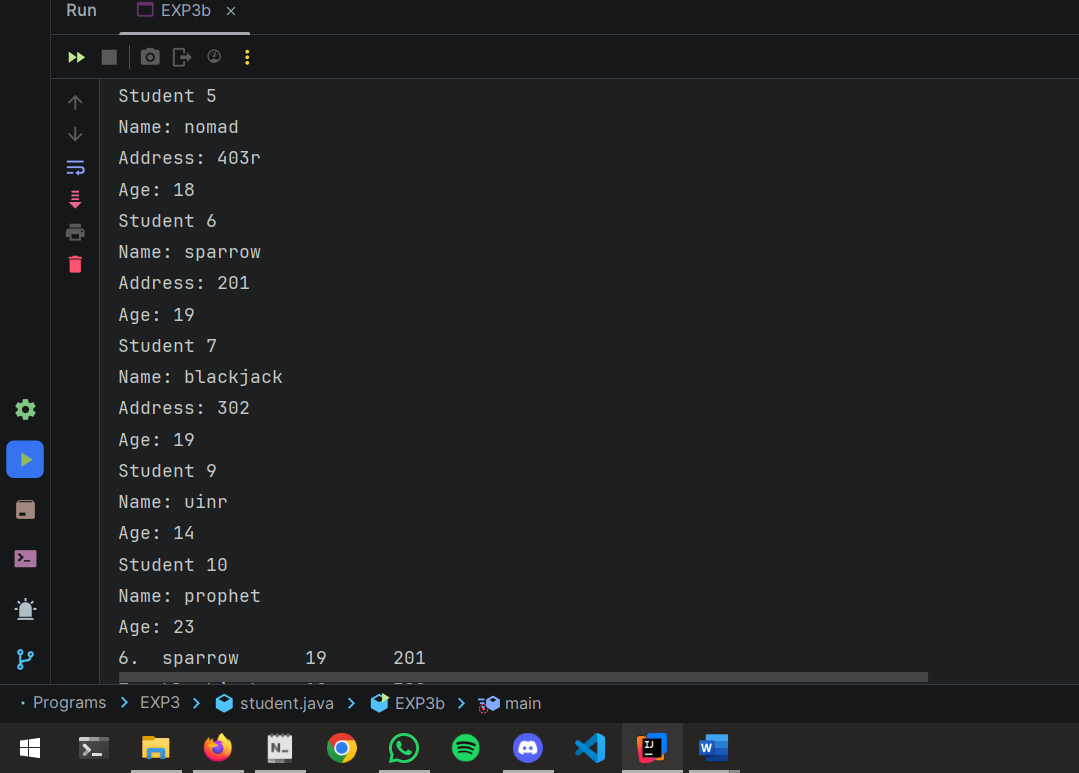
2) Implementation of Student class using default and parameterized Constructor and setInfo() default and parameterized method.

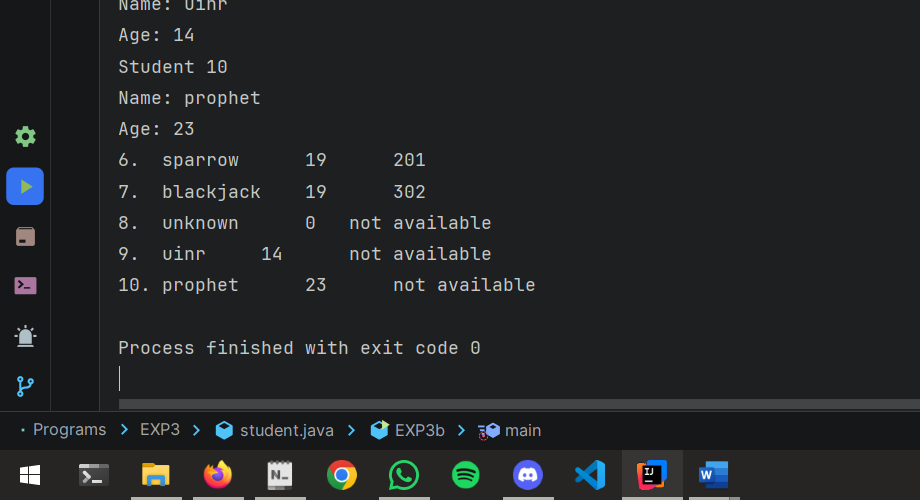
CODE:

package EXP3;  
  
import java.util.Scanner;  
  
class student {  
 int age;  
 String name, address;  
  
 public void setInfo(String name, int age) {  
 this.name = name;  
 this.age = age;  
 }  
  
 public void setInfo(String name, int age, String address) {  
 this.name = name;  
 this.age = age;  
 this.address = address;  
 }  
  
 public void showInfo() {  
 System.out.println(name + " \t " + age + " \t " + address);  
 }  
  
 student() {  
 this.name = "unknown";  
 this.age = 0;  
 this.address = "not available";  
 }  
  
 student(String name, int age, String address) {  
 this.name = name;  
 this.age = age;  
 this.address = address;  
 }  
}  
  
class EXP3b {  
 static Scanner sc = new Scanner(System.in);  
  
 public static void main(String[] args) {  
 student[] s = new student[10];  
 System.out.println("Please enter the details of the students:");  
 for (int i = 0; i < 5; i++) {  
 System.out.println("Student " + (i + 1));  
 System.out.print("Name: ");  
 String name = sc.next();  
 System.out.print("Address: ");  
 String address = sc.next();  
 System.out.print("Age: ");  
 int age = sc.nextInt();  
 s[i] = new student(name, age, address);  
 }  
  
 for (int i = 5; i < s.length; i++) {  
 s[i] = new student();  
 }  
 for (int i = 5; i < 7; i++) {  
 System.out.println("Student " + (i + 1));  
 System.out.print("Name: ");  
 String name = sc.next();  
 System.out.print("Address: ");  
 String address = sc.next();  
 System.out.print("Age: ");  
 int age = sc.nextInt();  
 s[i].setInfo(name, age, address);  
 }  
 for (int i = 8; i < s.length; i++) {  
 System.out.println("Student " + (i + 1));  
 System.out.print("Name: ");  
 String name = sc.next();  
 System.out.print("Age: ");  
 int age = sc.nextInt();  
 s[i].setInfo(name, age);  
 }  
 for (int i = 5; i < s.length; i++) {  
 System.out.print((i + 1) + ".\t");  
 s[i].showInfo();  
 }  
 }  
}

OUPTUT:







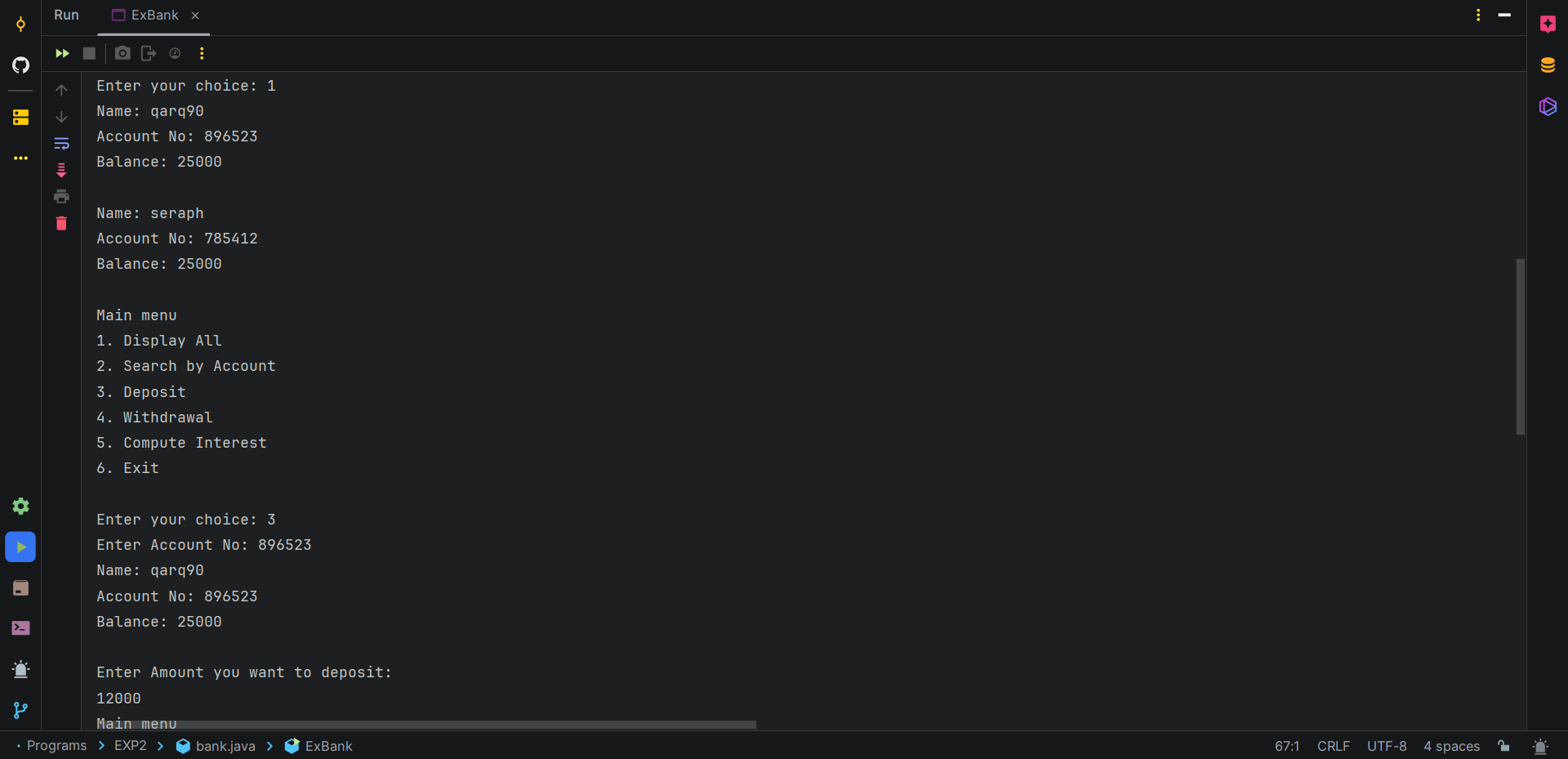
3) Implementation of Bank Example in Java using OOP.

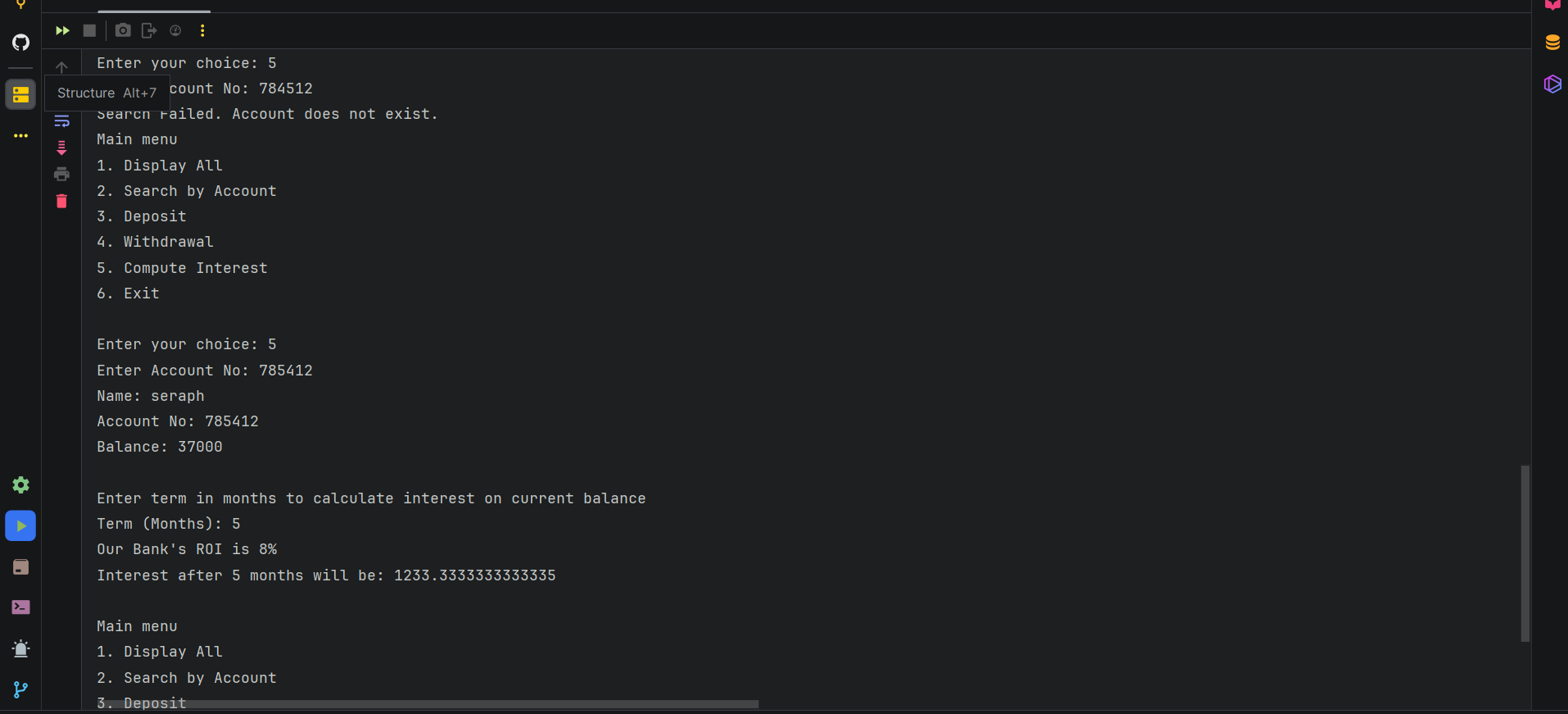
CODE:

package EXP2;  
  
import java.util.Scanner;  
  
class bank {  
 private String name;  
 private String accno;  
 private static long balance = 0;  
  
 static Scanner KB = new Scanner(System.in);  
  
 void createAccount() {  
 System.out.println("Enter Name: ");  
 name = KB.next();  
 System.out.println("Enter Account No: ");  
 accno = KB.next();  
 System.out.println("Enter Balance: ");  
 balance = KB.nextLong();  
 System.out.println("\n");  
 }  
  
 void showAccount() {  
 System.out.println("Name: " + name + "\nAccount No: " + accno + "\nBalance: " + balance + "\n");  
 }  
  
 void deposit() {  
 long amt;  
 System.out.println("Enter Amount you want to deposit: ");  
 amt = KB.nextLong();  
 balance += amt;  
 }  
  
 void withdraw() {  
 long amt;  
 System.out.println("Enter amount you want to withdraw: ");  
 amt = KB.nextLong();  
 if (balance >= amt) {  
 balance -= amt;  
 } else {  
 System.out.println("Less balance. Transaction failed.");  
 }  
 }  
  
 boolean search(String acn) {  
 if (accno.equals(acn)) {  
 showAccount();  
 return true;  
 }  
 return false;  
 }  
  
 void computeInterest() {  
 System.out.println("Enter term in months to calculate interest on current balance");  
 System.out.print("Term (Months): ");  
 int t = KB.nextInt();  
 System.out.println("Our Bank's ROI is 8%");  
 double interest = getBalance() \* 0.08 \* (t / 12.0);  
 System.out.println("Interest after " + t + " months will be: " + interest);  
 System.out.println("");  
 }  
  
 static double getBalance() {  
 return balance;  
 }  
}  
  
class ExBank {  
 public static void main(String[] args) {  
 Scanner KB = new Scanner(System.in);  
 System.out.println("How many customers do you want to input? ");  
 int n = KB.nextInt();  
 bank[] C = new bank[n];  
 for (int i = 0; i < C.length; i++) {  
 C[i] = new bank();  
 C[i].createAccount();  
 }  
 int ch;  
 do {  
 System.out.println("Main menu\n1. Display All\n2. Search by Account\n3. Deposit\n4. Withdrawal\n5. Compute Interest\n6. Exit ");  
 System.out.print("\nEnter your choice: ");  
 ch = KB.nextInt();  
 switch (ch) {  
 case 1:  
 for (bank bank : C) {  
 bank.showAccount();  
 }  
 break;  
 case 2:  
 System.out.print("Enter Account No you want to search: ");  
 String acn = KB.next();  
 boolean found = false;  
 for (bank bank : C) {  
 found = bank.search(acn);  
 if (found) {  
 break;  
 }  
 }  
 if (!found) {  
 System.out.println("Search Failed. Account does not exist.");  
 }  
 break;  
 case 3:  
 System.out.print("Enter Account No: ");  
 acn = KB.next();  
 found = false;  
 for (bank bank : C) {  
 found = bank.search(acn);  
 if (found) {  
 bank.deposit();  
 break;  
 }  
 }  
 if (!found) {  
 System.out.println("Search Failed. Account does not exist.");  
 }  
 break;  
 case 4:  
 System.out.print("Enter Account No: ");  
 acn = KB.next();  
 found = false;  
 for (bank bank : C) {  
 found = bank.search(acn);  
 if (found) {  
 bank.withdraw();  
 break;  
 }  
 }  
 if (!found) {  
 System.out.println("Search Failed. Account does not exist.");  
 }  
 break;  
 case 5:  
 System.out.print("Enter Account No: ");  
 acn = KB.next();  
 found = false;  
 for (bank bank : C) {  
 found = bank.search(acn);  
 if (found) {  
 bank.computeInterest();  
 break;  
 }  
 }  
 if (!found) {  
 System.out.println("Search Failed. Account does not exist.");  
 }  
 break;  
 case 6:  
 System.out.println("Exiting...");  
 break;  
 default:  
 System.out.println("Invalid choice.");  
 break;  
 }  
 } while (ch != 6);  
 KB.close();  
 }  
}

OUTPUT:









4) Implementation of MCQ Application in java.

CODE:

package EXP3;  
  
import java.util.Scanner;  
  
class User {  
 int score = 0, id;  
 String name;  
  
 public void showResult() {  
 System.out.println(id + "\t" + name + "\t" + score);  
 }  
  
 User(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
}  
  
class MCQ {  
 String question;  
 String[] option;  
 String topic;  
 int answer;  
  
 public void setTopic(String topic) {  
 this.topic = topic;  
 }  
  
 public void setQuestion(String question) {  
 this.question = question;  
 }  
  
 public void setOption(String[] option) {  
 this.option = option;  
 }  
  
 public void setAnswer(int answer) {  
 this.answer = answer;  
 }  
  
 public void dispMCQ() {  
 System.out.println(question);  
 for (int i = 0; i < 4; i++) {  
 System.out.print((i + 1) + ". " + option[i] + "\t");  
 }  
 System.out.println("\n");  
 }  
  
 MCQ(String topic, String question, String[] option, int answer) {  
 this.topic = topic;  
 this.question = question;  
 this.option = option;  
 this.answer = answer;  
 }  
}  
  
class EXP3d {  
 static Scanner sc = new Scanner(System.in);  
  
 public static void main(String[] args) {  
 String topic1 = "Java";  
 String[] question1 = {  
 "Which of these is not a feature of Java?",  
 "Which component of Java is responsible for running the compiled Java bytecode?",  
 "Which feature of Java makes it possible to run a Java program on different platforms?",  
 "In Java, how should class names be written?",  
 "Which keyword is used to define a constant variable in Java?"  
 };  
 String[][] options1 = {  
 {"Object-oriented", "Platform-independent", "Compiled", "Interpreted Language"},  
 {"JDK", "JVM", "JRE", "JIT"},  
 {"Object-oriented", "Platform-independent", "Syntax", "Memory Management"},  
 {"camelCase", "snake\_case", "PascalCase", "kebab-case"},  
 {"final", "static", "const", "immutable"}  
 };  
 int[] answers1 = {3, 2, 2, 3, 1};  
  
 String topic2 = "C++";  
 String[] question2 = {  
 "Identify the incorrect constructor type",  
 "Identify the logical AND operator.",  
 "Under which pillar of OOPS does base class and derived class relationship come?",  
 "Using which of the following keywords can an exception be generated?",  
 "What does a C++ class hold?"  
 };  
 String[][] options2 = {  
 {"Friend constructor", "Default constructor", "Parameterized constructor", "Copy constructor"},  
 {"||", "&&", "&", "!"},  
 {"Polymorphism", "Inheritance", "Encapsulation", "Abstraction"},  
 {"threw", "throws", "throw", "catch"},  
 {"Function", "Data", "Arrays", "Both a and b"}  
 };  
 int[] answers2 = {1, 2, 2, 3, 4};  
  
 MCQ[] mcqJava = new MCQ[5];  
 MCQ[] mcqCpp = new MCQ[5];  
  
 for (int i = 0; i < mcqJava.length; i++) {  
 mcqJava[i] = new MCQ(topic1, question1[i], options1[i], answers1[i]);  
 }  
  
 for (int i = 0; i < mcqCpp.length; i++) {  
 mcqCpp[i] = new MCQ(topic2, question2[i], options2[i], answers2[i]);  
 }  
  
 System.out.print("Please enter your ID: ");  
 int id = sc.nextInt();  
 sc.nextLine();  
  
 System.out.print("Please enter your name: ");  
 String name = sc.nextLine();  
  
 System.out.println("Please choose your test topic: \n1. Java \t 2. C++");  
 int choice = sc.nextInt();  
  
 User user = new User(id, name);

switch (choice) {  
 case 1:  
 for (int i = 0; i < mcqJava.length; i++) {  
 System.out.print("\n" + (i + 1) + ". ");  
 mcqJava[i].dispMCQ();  
 System.out.print("Your answer: ");  
 int answer = sc.nextInt();  
 if (answer == mcqJava[i].answer) {  
 user.score++;  
 }  
 }  
 break;  
  
 case 2:  
 for (int i = 0; i < mcqCpp.length; i++) {  
 System.out.print("\n" + (i + 1) + ". ");  
 mcqCpp[i].dispMCQ();  
 System.out.print("Your answer: ");  
 int answer = sc.nextInt();  
 if (answer == mcqCpp[i].answer) {  
 user.score++;  
 }  
 }  
 break;  
  
 default:  
 System.out.println("Invalid choice entered!\nExiting...");  
 break;  
 }  
  
 System.out.println("\nID\tName\tScore");  
 user.showResult();  
 }  
}

OUTPUT:

