www.ignousite.com
Course Code : BCSL-043
Title : Java Programming Lab
Assignment Number : BCA(4)/L-043/Assignment/2022-23
Maximum Marks : 50 www.ignousite.com
Last date of Submission : 31st October, 2022 (for July Session)
: 15th April, 2023 (for January Session)

Question 1 (a): Write a java program for Matrix Multiplication. Make necessary assumptions.

Ans. // Java program to multiply two square matrices.

```
java.io.*;

class GFG {

	// Function to print Matrix
	static void printMatrix(int M[][],
							int rowSize,
							int colSize)
	{
		for (int i = 0; i < rowSize; i++) {
			for (int j = 0; j < colSize; j++)
				System.out.print(M[i][j] + " ");

			System.out.println();
		}
	}

	// Function to multiply
	// two matrices A[][] and B[][]
	static void multiplyMatrix(
		int row1, int col1, int A[][],
		int row2, int col2, int B[][])
	{
		int i, j, k;

		// Print the matrices A and B
		System.out.println("\nMatrix A:");
		printMatrix(A, row1, col1);
		System.out.println("\nMatrix B:");
		printMatrix(B, row2, col2);

		// Check if multiplication is Possible
		if (row2 != col1) {

			System.out.println(
				"\nMultiplication Not Possible");
			return;
```

```
        }

        // Matrix to store the result
        // The product matrix will
        // be of size row1 x col2
        int C[][] = new int[row1][col2];

        // Multiply the two matrices
        for (i = 0; i < row1; i++) {
            for (j = 0; j < col2; j++) {
                for (k = 0; k < row2; k++)
                    C[i][j] += A[i][k] * B[k][j];
            }
        }

        // Print the result
        System.out.println("\nResultant Matrix:");
        printMatrix(C, row1, col2);
    }

    // Driver code
    public static void main(String[] args)
    {

        int row1 = 4, col1 = 3, row2 = 3, col2 = 4;

        int A[][] = { { 1, 1, 1 },
                      { 2, 2, 2 },
                      { 3, 3, 3 },
                      { 4, 4, 4 } };

        int B[][] = { { 1, 1, 1, 1 },
                      { 2, 2, 2, 2 },
                      { 3, 3, 3, 3 } };

        multiplyMatrix(row1, col1, A,
                      row2, col2, B);
    }
}
```

Output:

```
Matrix A:
1  1  1
2  2  2
3  3  3
4  4  4
```

```
Matrix B:
1  1  1  1
2  2  2  2
3  3  3  3

Resultant Matrix:
 6  6  6  6
12 12 12 12
18 18 18 18
24 24 24 24
```

(b) Write a Java program to define Book class and appropriate constructor for the class. Define proper getter and sett methods for the class. Make necessary assumptions.

Ans.

```
class Book
{
  // class member variable
  private int bId;
  private String bName;
  private String bDesignation;
  private String bAuthor;

  public int getbookId()
  {
    return bId;
  }
  public void setbookId(final int bId)
  {
    this.bId = bId;
  }
  public String getbookName()
  {
    return bName;
  }
  public void setbookName(final String bName)
  {
    // Validating the book's name and
    // throwing an exception if the name is null or its length is less than or equal to 0.
    if(bName == null ||  bName.length() <= 0)
    {
      throw new IllegalArgumentException();
    }
    this.bName = bName;
  }
```

```
    public String getbookDesignation()
    {
        return bDesignation;
    }
    public void setbookDesignation(final String bDesignation)
    {
        this.bDesignation = bDesignation;
    }
    public String getbookAuthor()
    {
        return bAuthor;
    }

    public void setbookAuthor (final String bAuthor)
    {
        this.bAuthor = bAuthor;
    }
    // for printing the values
    @Override
    public String toString()
    {
        String str = "Book: [id = " + getbookId() + ", name = " + getbookName() + ", designation = " + getbookDesignation() + ", Autho
r = " + getbookAuthor () + "]";
        return str;
    }
}
// Main class.
public class GetterSetterExample1
{
    // main method
    public static void main(String argvs[])
    {
        // Creating an object of the Book class
        final Book book = new Book();

        // the book details are getting set using the setter methods.
        emp.setbookId(107);
        emp.setbookName("Java Programming Lab");
        emp.setbookDesignation("Software");
        emp.setbookAuthor("Ignou Study Helper Corporation");

        // Displaying the details of the book details using the
        // 'toString()' method, which uses the getter methods
        System.out.println(book.toString());
    }
```

```
}
```

Output:

```
Book: [id = 107, name = Java Programming Lab, designation = Software, company = Ignou Study Helper Corporation
```

(c) Write a program to demonstrate use of:

i. Multithreading

ii. Exceptions Handling

Ans. i. Multithreading

```
// Java code for thread creation by extending
// the Thread class
class MultithreadingDemo extends Thread {
        public void run()
        {
                try {
                        // Displaying the thread that is running
                        System.out.println(
                                "Thread " + Thread.currentThread().getId()
                                + " is running");
                }
                catch (Exception e) {
                        // Throwing an exception
                        System.out.println("Exception is caught");
                }
        }
}

// Main Class
public class Multithread {
        public static void main(String[] args)
        {
                int n = 8; // Number of threads
                for (int i = 0; i < n; i++) {
                        MultithreadingDemo object
                                = new MultithreadingDemo();
                        object.start();
                }
        }
}
```

Output:

```
Thread 15 is running
Thread 14 is running
Thread 16 is running
Thread 12 is running
Thread 11 is running
Thread 13 is running
Thread 18 is running
Thread 17 is running
```

ii. Exceptions Handling

```
class Main {
  public static void main(String[] args) {
    try {
      // code that generates exception
      int divideByZero = 5 / 0;
    }

    catch (ArithmeticException e) {
      System.out.println("ArithmeticException => " + e.getMessage());
    }

    finally {
      System.out.println("This is the finally block");
    }
  }
}
```

Output:

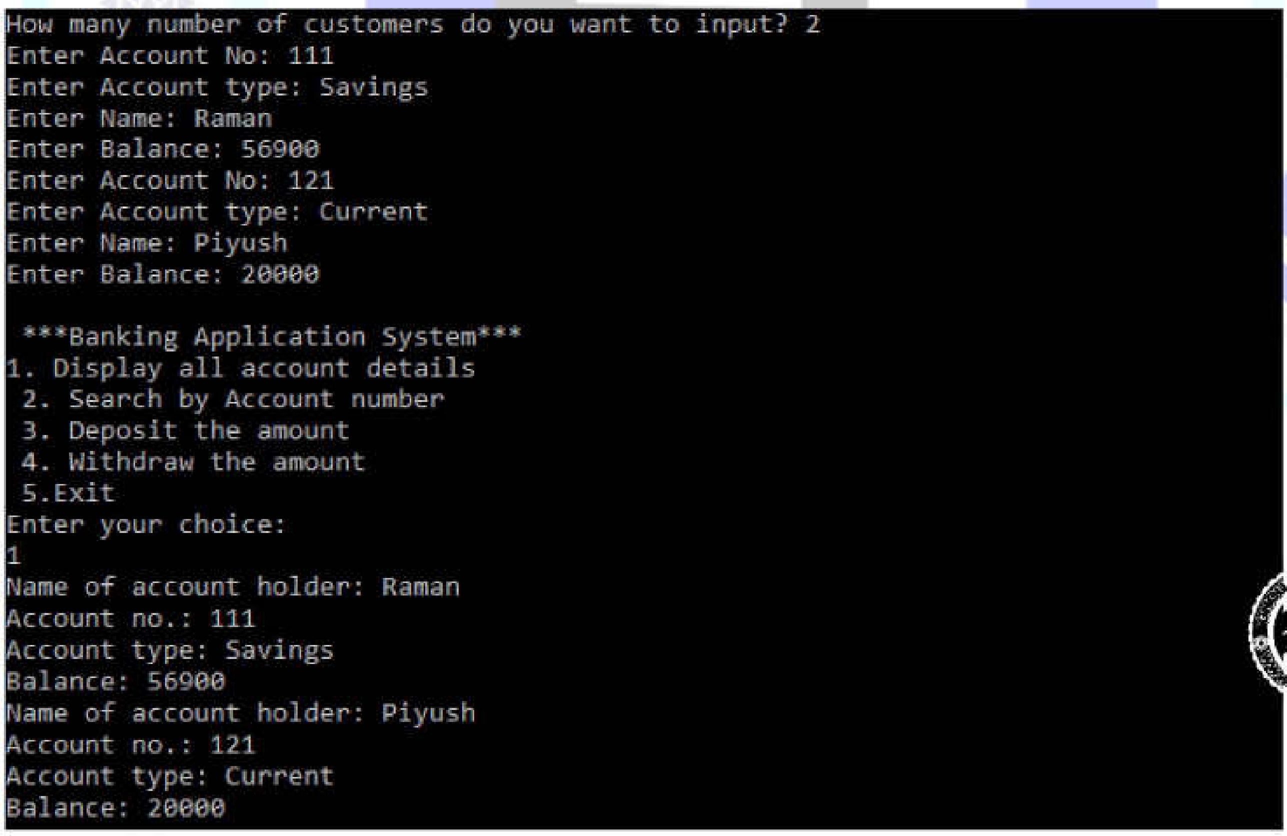
```
ArithmeticException => / by zero
This is the finally block
```

Q2. (a) Write a program in Java which define an abstract class BankAccount. Using this class define some concrete classes. Make necessary assumptions.

Ans.

```
import java.util.Scanner;
abstract class BankAccount {
  private String accno;
  private String name;
  private String acc_type;
  private long balance;
  Scanner sc = new Scanner(System.in);
  //method to open new account
  public void openAccount() {
    System.out.print("Enter Account No: ");
    accno = sc.next();
    System.out.print("Enter Account type: ");
    acc_type = sc.next();
    System.out.print("Enter Name: ");
    name = sc.next();
    System.out.print("Enter Balance: ");
    balance = sc.nextLong();
    }
    //method to display account details
    public void showAccount() {
        System.out.println("Name of account holder: " + name);
        System.out.println("Account no.: " + accno);
        System.out.println("Account type: " + acc_type);
        System.out.println("Balance: " + balance);
    }
    //method to deposit money
    public void deposit() {
        long amt;
        System.out.println("Enter the amount you want to deposit: ");
        amt = sc.nextLong();
        balance = balance + amt;
    }
    //method to withdraw money
    public void withdrawal() {
        long amt;
        System.out.println("Enter the amount you want to withdraw: ");
        amt = sc.nextLong();
        if (balance >= amt) {
            balance = balance - amt;
            System.out.println("Balance after withdrawal: " + balance);
        } else {
            System.out.println("Your balance is less than " + amt + "\tTransaction failed...!!" );
        }
    }
    //method to search an account number
    public boolean search(String ac_no) {
        if (accno.equals(ac_no)) {
            showAccount();
            return (true);
        }
        return (false);
    }
}
public class BankingApp {
    public static void main(String arg[]) {
        Scanner sc = new Scanner(System.in);
        //create initial accounts
        System.out.print("How many number of customers do you want to input? ");
        int n = sc.nextInt();
        BankDetails C[] = new BankDetails[n];
        for (int i = 0; i < C.length; i++) {
            C[i] = new BankDetails();
        C[i].openAccount();
      }
      // loop runs until number 5 is not pressed to exit
      int ch;
      do {
        System.out.println("\n ***Banking System Application***");
        System.out.println("1. Display all account details \n 2. Search by Account number\n 3. Deposit the amount \n 4. Withdraw
the amount \n 5.Exit ");
        System.out.println("Enter your choice: ");
        ch = sc.nextInt();
          switch (ch) {
            case 1:
              for (int i = 0; i < C.length; i++) {
                C[i].showAccount();
              }
              break;
            case 2:
              System.out.print("Enter account no. you want to search: ");
              String ac_no = sc.next();
              boolean found = false;
              for (int i = 0; i < C.length; i++) {
                found = C[i].search(ac_no);
                if (found) {
                  break;
                }
              }
              if (!found) {
                System.out.println("Search failed! Account doesn't exist..!!");
              }
              break;
            case 3:
              System.out.print("Enter Account no. : ");
              ac_no = sc.next();
              found = false;
              for (int i = 0; i < C.length; i++) {
                found = C[i].search(ac_no);
                if (found) {
                  C[i].deposit();
                  break;
                }
              }
              if (!found) {
                System.out.println("Search failed! Account doesn't exist..!!");
              }
              break;
            case 4:
              System.out.print("Enter Account No : ");
              ac_no = sc.next();
              found = false;
              for (int i = 0; i < C.length; i++) {
                found = C[i].search(ac_no);
                if (found) {
                  C[i].withdrawal();
                  break;
                }
              }
              if (!found) {
                System.out.println("Search failed! Account doesn't exist..!!");
              }
              break;
            case 5:
              System.out.println("See you soon...");
              break;
          }
        }
        while (ch != 5);
    }
}
```

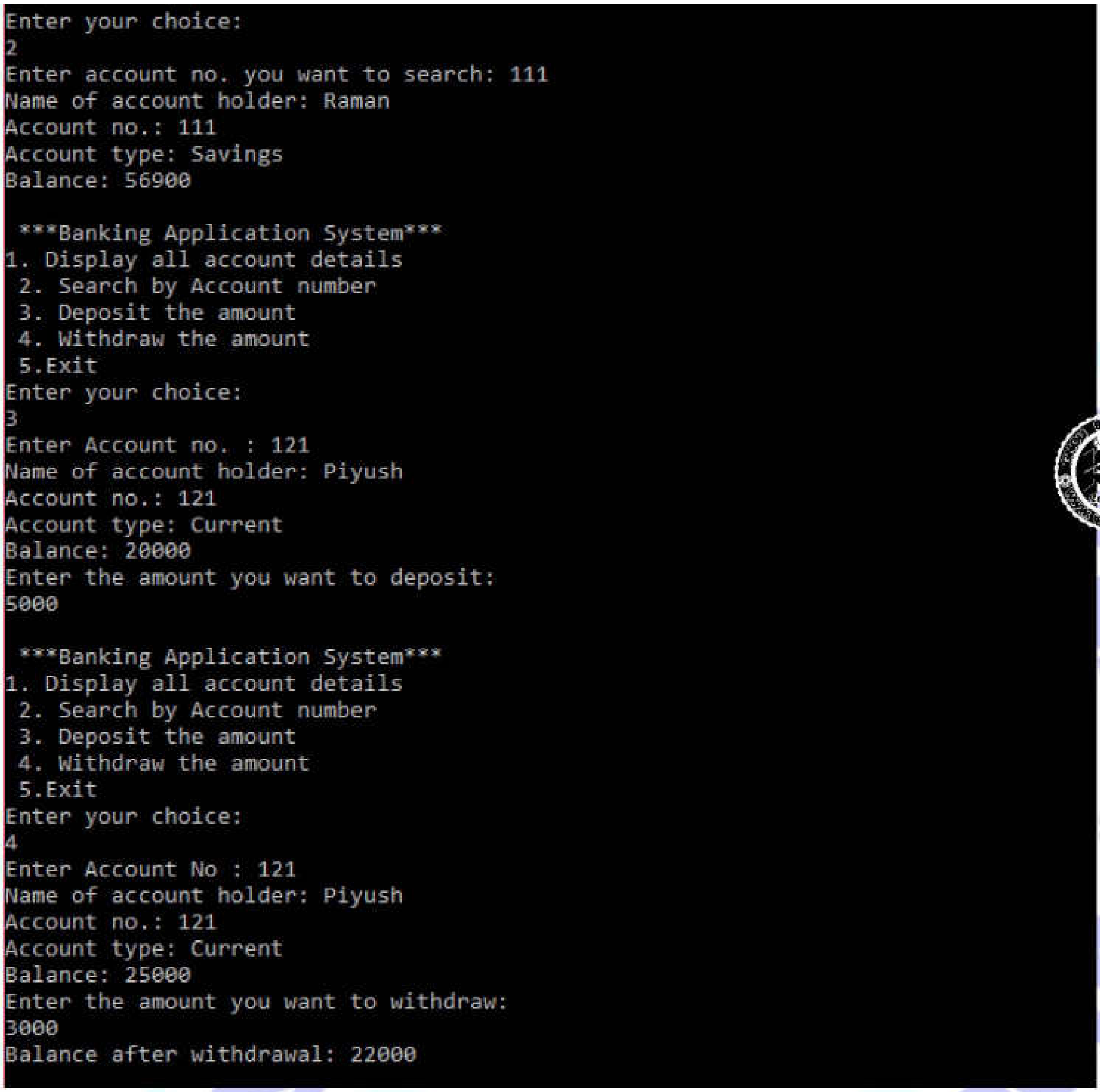
Output 1:



```
How many number of customers do you want to input? 2
Enter Account No: 111
Enter Account type: Savings
Enter Name: Raman
Enter Balance: 56900
Enter Account No: 121
Enter Account type: Current
Enter Name: Piyush
Enter Balance: 20000

 ***Banking Application System***
1. Display all account details
 2. Search by Account number
 3. Deposit the amount
 4. Withdraw the amount
 5.Exit
Enter your choice:
1
Name of account holder: Raman
Account no.: 111
Account type: Savings
Balance: 56900
Name of account holder: Piyush
Account no.: 121
Account type: Current
Balance: 20000
```

Output 2:



```
Enter your choice:
2
Enter account no. you want to search: 111
Name of account holder: Raman
Account no.: 111
Account type: Savings
Balance: 56900

 ***Banking Application System***
1. Display all account details
 2. Search by Account number
 3. Deposit the amount
 4. Withdraw the amount
 5.Exit
Enter your choice:
3
Enter Account no. : 121
Name of account holder: Piyush
Account no.: 121
Account type: Current
Balance: 20000
Enter the amount you want to deposit:
5000

 ***Banking Application System***
1. Display all account details
 2. Search by Account number
 3. Deposit the amount
 4. Withdraw the amount
 5.Exit
Enter your choice:
4
Enter Account No : 121
Name of account holder: Piyush
Account no.: 121
Account type: Current
Balance: 25000
Enter the amount you want to withdraw:
3000
Balance after withdrawal: 22000
```

(b) Write a program in Java to create an applet which draw either a rectangle or a circle on the basis of choice of input.

Ans. Our choice are rectangle:

```
// Java Program to Draw a rectangle

// using drawRect(int x, int y, int width, int height)

import java.awt.*;

import javax.swing.*;
```

```
public class rectangle extends JApplet {

        public void init()

        {

                // set size

                setSize(400, 400);


                repaint();

        }


        // paint the applet

        public void paint(Graphics g)

        {

                // set Color for rectangle

                g.setColor(Color.red);


                // draw a rectangle

                g.drawRect(100, 100, 200, 200);

        }

}
```

Output: