

ASSIGNMENT – 7

- 1. Write an exception handling java program to read two numbers n1, n2 and calculate and print the result of n1/n2. If n2 is Zero (0) then it will be handled by exception handler and again ask the value of n2. In the exception handler the program should display appropriate message to the user.**

```
import java.util.Scanner;
class DivExcep {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        boolean valid = false;
        System.out.print("Enter the first number: ");
        int n1 = sc.nextInt();
        System.out.print("Enter the second number: ");
        int n2 = sc.nextInt();
        try {
            int res = n1 / n2;
            System.out.println("Result is: "+res);
        }
        catch (ArithmaticException e) {
            System.out.println("MathError: Can't be divide by 0!");
        }
    }
}
```

```
C:\24BCSA08\assignment7>java q1.java
Enter the first number: 5
Enter the second number: 0
MathError: Can't be divide by 0!
```

- 2. Write a java program to read two numbers x and y and calculate x/(x-y). The program should check the value of x-y. Before dividing with x, it should throw an exception if x-y is zero. In the exception handler the program should display appropriate message to the user.**

```
import java.util.Scanner;
class DivCheck {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int n1 = sc.nextInt();
        System.out.print("Enter the second number: ");
        int n2 = sc.nextInt();

        try {
            double res = n1 / (n1-n2);
            System.out.println("Result: "+res);
        }
    }
}
```

```

        catch (ArithmaticException e) {
            System.out.println("Error: "+e+" Can't divide by 0!");
        }
    }
}

```

```

C:\24BCSA08\assignment7>java q2.java
Enter the first number: 4
Enter the second number: 4
Error: java.lang.ArithmaticException: / by zero Can't divide by 0!

```

3. Write an exception handling java program to print the index position of an existing integer array. The index value will be entered by user. It will be handled by exception handler if index position is greaterthan the size of array. In the exception handler the program should display appropriate message to the user.

```
import java.util.Scanner;
```

```

public class ArrayIndexExcep {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the size of the array: ");
        int size = sc.nextInt();
        int[] arr = new int[size];

        System.out.println("Enter " + size + " elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.println("Array elements are: ");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }

        try {
            System.out.print("\nEnter the index position to access: ");
            int index = sc.nextInt();
            System.out.println("Element at index " + index + " is: " + arr[index]);
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Invalid index. Please enter an index
between 0 and " + (arr.length - 1) + ".");
        }
    }
}

```

```
C:\24BCSA08\assignment7>java q3.java
Enter the size of the array: 5
Enter 5 elements:
2
2
2
2
2
Array elements are:
2 2 2 2 2
Enter the index position to access: 6
Error: Invalid index. Please enter an index between 0 and 4.
```

4. Write a program to illustrate the use of multiple catch blocks associated with a single try block.

```
import java.util.Scanner;
```

```
public class MultiCatch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            System.out.print("Enter the first number: ");
            int a = sc.nextInt();
            System.out.print("Enter the second number: ");
            int b = sc.nextInt();

            int result = a / b;
            System.out.println("Result of division: " + result);

            int[] arr = {10, 20, 30};
            System.out.print("Enter an array index to access: ");
            int index = sc.nextInt();
            System.out.println("Element at index " + index + ": " + arr[index]);
        }
        catch (ArithmException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Invalid array index.");
        }
    }
}
```

```
C:\24BCSA08\assignment7>java q4.java
Enter the first number:
C:\24BCSA08\assignment7>java q4.java
Enter the first number: 4
Enter the second number: 4
Result of division: 1
Enter an array index to access: 4
Error: Invalid array index.
```

5. Write a class called Account with the following properties and methods:

Properties: String name, int acc_no, double balance, Methods: void deposit (double amt), void withdraw (double amt).

Assume that an account needs to have a minimum balance of 500. If an attempt is made to withdraw, which results in balance going below 500, throw a user defined exception called MinimumBalanceException. Use throw and throws wherever necessary.

```
import java.util.Scanner;
```

```
class MinimumBalanceException extends Exception {
    public MinimumBalanceException(String message) {
        super(message);
    }
}

class Account {
    String name;
    int acc_no;
    double balance;

    Account(String name, int acc_no, double balance) {
        this.name = name;
        this.acc_no = acc_no;
        this.balance = balance;
    }

    void deposit(double amt) {
        balance += amt;
        System.out.println("Amount deposited successfully. Current balance: " + balance);
    }

    void withdraw(double amt) throws MinimumBalanceException {
        if (balance - amt < 500) {
            throw new MinimumBalanceException("Withdrawal denied. Minimum balance of 500 must be maintained.");
        } else {
            balance -= amt;
            System.out.println("Amount withdrawn successfully. Current balance: " + balance);
        }
    }
}
```

```

public class BankDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter account holder name: ");
        String name = sc.nextLine();
        System.out.print("Enter account number: ");
        int acc_no = sc.nextInt();
        System.out.print("Enter initial balance: ");
        double balance = sc.nextDouble();

        Account acc = new Account(name, acc_no, balance);

        System.out.print("Enter amount to deposit: ");
        double dep = sc.nextDouble();
        acc.deposit(dep);

        System.out.print("Enter amount to withdraw: ");
        double wd = sc.nextDouble();

        try {
            acc.withdraw(wd);
        } catch (MinimumBalanceException e) {
            System.out.println("Error: " + e.getMessage());
        }

        sc.close();
    }
}

```

```

C:\24BCSJ18>javac BankDemo.java

C:\24BCSJ18>java BankDemo
Enter account holder name: SRIYAS
Enter account number: 234
Enter initial balance: 34555
Enter amount to deposit: 3455
Amount deposited successfully. Current balance: 38010.0
Enter amount to withdraw: 1200000
Error: Withdrawal denied. Minimum balance of 500 must be maintained.

C:\24BCSJ18>

```

6. Write a program that prompts the user to enter a length in feet and inches and outputs the equivalent length in centimetres. If the user enters a negative number or a non-digit number, throw and handle an appropriate exception and prompt the user to enter another set of numbers.

```
import java.util.Scanner;
```

```
public class MeasureExcep {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int feet = 0, inches = 0;  
        boolean valid = false;  
  
        while (!valid) {  
            try {  
                System.out.print("Enter feet: ");  
                String feetInput = sc.next();  
                feet = Integer.parseInt(feetInput);  
                if (feet < 0) throw new NumberFormatException("Negative number not allowed.");  
  
                System.out.print("Enter inches: ");  
                String inchInput = sc.next();  
                inches = Integer.parseInt(inchInput);  
                if (inches < 0) throw new NumberFormatException("Negative number not  
allowed.");  
  
                valid = true;  
            } catch (NumberFormatException e) {  
                System.out.println("Invalid input: " + e.getMessage() + " Please enter non-negative  
integers.");  
            }  
        }  
  
        double totalInches = feet * 12 + inches;  
        double cm = totalInches * 2.54;  
        System.out.println("Equivalent length in centimetres: " + cm);  
  
        sc.close();  
    }  
}
```

```
C:\24BCSA08\assignment7>java q6.java  
Enter feet:  
-5  
Invalid input: Negative number not allowed. Please enter non-negative integers.  
Enter feet: 4  
Enter inches: 9  
Equivalent length in centimetres: 144.78
```