

ASSIGNMENT 09

1) Write a Java program to read two numbers a and b and calculate $a/(a-b)$. The program should check the value of a-b before dividing with a, it should throw an exception if a-b is zero. In the exception handler the program should display appropriate message to the user.

```
import java.util.Scanner;

public class Ex1 {
    public static void main(String[] args) {
        int a, b;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter two numbers");
        a = in.nextInt();
        b = in.nextInt();
        int r;
        int d = a-b;
        try{
            r = a/d;
            System.out.println("Result of a/a-b = "+r);
        }
        catch(ArithmeticException e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
Enter two numbers
2
2
java.lang.ArithmeticException: / by zero
```

2) Write a class Account with the following properties and methods: Properties: String name, int acc_no, double balance Methods: void deposit(int num), void withdraw(int num), void transfer(Account acc1, Account acc2, int amt) Assume that an account needs to have a minimum balance of 500. If an attempt is made to withdraw or transfer, which results in balance going below 500, throw a user-defined exception called MinmunBalanceException. Use throw and throw wherever necessary

```
import java.util.Scanner;

class MinBalEx extends Exception {
    String str;

    MinBalEx(String str) {
        this.str = str;
    }

    public String toString() {
        return str;
    }
}
```

```

    }
}

class Account {
    String name;
    int acc_no;
    double balance = 500;

    void deposit(int num) {
        balance = balance + num;
        System.out.println("After deposite the account balance is: " + balance);
    }

    void withdraw(int num) throws MinBalEx {
        double d = balance - num;
        if (d <= 500) {
            throw new MinBalEx("MinimunBalanceException");
        } else {
            balance = balance - num;
            System.out.println("After withdral your account balance is: " + balance);
        }
    }

    void transfer(Account acc1, Account acc2, int amt) throws MinBalEx {
        double d = acc1.balance - amt;
        if (d <= 500) {
            throw new MinBalEx("MinimunBalanceException");
        } else {
            acc1.balance = acc1.balance - amt;
            System.out.println("Your trasaction is successfull, After tranfer your accont balance is: " +
acc1.balance);
        }
        acc2.balance = acc2.balance + amt;
        System.out.println(
            "Your bank account has credited by " + amt + " rupees, Now your bank balance is: " +
acc2.balance);
    }
}

public class Bank {
    public static void main(String[] args) throws MinBalEx {
        Scanner in = new Scanner(System.in);
        Account ac1 = new Account();
        Account ac2 = new Account();
        Account ac3 = new Account();
        int ch;
        int f = 0;
        while (f==0) {
            System.out.println("Enter your choice");
            System.out.println("0. Terminate the transaction");
            System.out.println("1. Deposite");
            System.out.println("2. Withdrawal");
            System.out.println("3. Transfer");
            ch = in.nextInt();

            switch (ch) {
                case 0:
                    f=1;
                    break;
                case 1:
                    int amt;
                    System.out.println("Enter the amount to deposit");
                    amt = in.nextInt();
                    ac1.deposit(amt);
                    break;
                case 2:
                    int am;
                    System.out.println("Enter the amount to withdraw");

```

```

        am = in.nextInt();
        try{
            ac1.withdraw(am);
        }
        catch(MinBalEx e){
            System.out.println(e);
        }
        break;
    case 3:
        int amnt;
        System.out.println("Enter the amount to tranfer");
        amnt = in.nextInt();
        ac3.transfer(ac1, ac2, amnt);
        break;
    default:
        System.out.println("Invalid Choice");
        break;
    }
}
}
}
}

```

OUTPUT:

Enter your choice

0. Terminate the transaction

1. Deposit

2. Withdrawal

3. Transfer

1

Enter the amount to deposit

2500

After deposit the account balance is: 3000.0

Enter your choice

0. Terminate the transaction

1. Deposit

2. Withdrawal

3. Transfer

2

Enter the amount to withdraw

2600

MinimunBalanceException

Enter your choice

0. Terminate the transaction

1. Deposit

2. Withdrawal

3. Transfer

3

Enter the amount to tranfer

1500

Your trasaction is successfull, After tranfer your accont balance is: 1500.0

Your bank account has credited by 1500 rupees, Now your bank balance is: 2000.0

Enter your choice

0. Terminate the transaction

1. Deposit

2. Withdrawal

3. Transfer

0

4) Write Java programs to implement all cases of exception handling:

a. ArithmeticException

b. ArrayIndexOutOfBoundsException

c. NullPointerException

d. NumberFormatException

```
// a. Arithmetic Exception
public class ExTy {
    public static void main(String[] args) {
        int no=10;
        try{
            int res = no/0;
            System.out.println(res);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

Java.lang.ArithmeticException

```
// b. ArrayIndexOutOfBoundsException
public class ExTy {
    public static void main(String[] args) {
        int arr[] = new int[1];
        try{
            System.out.println(arr[1]);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

java.lang.ArrayIndexOutOfBoundsException: Index 1 out of bounds for length 1

```
// c. NullPointerException
public class ExTy {
    public static void main(String[] args) {
        String str = null;
        try{
            int no = str.length();
            System.out.println(no);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
java.lang.NullPointerException: Cannot invoke "String.length()"
because "<local1>" is null
```

```
// d. NumberFormatException
public class ExTy {
    public static void main(String[] args) {
        String str = "A";
        try{
            int no = Integer.parseInt(str);
            System.out.println(no);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
java.lang.NumberFormatException: For input string: "A"
```

4) Write Java programs to implement exception handling by using:

- a. try block and multiple catch block**
- b. Nested try-catch block**
- c. finally block**
- d. throw keyword**

e. throws keyword

f. User-defined Exception/ Custom Exception

g. Generic Exception

a.

```
import java.util.Scanner;
public class ExTy {
    public static void main(String[] args) {
        int no1, no2, res;
        System.out.println("Welcome");
        Scanner in = new Scanner(System.in);
        try{
            System.out.println("Enter valu1");
            no1 = in.nextInt();
            System.out.println("Enter value");
            no2 = in.nextInt();
            res = no1/no2;
            System.out.println("Result is "+res);
        }
        catch(NullPointerException e){
            System.out.println(e);
        }
        catch(ArithmeticException e){
            System.out.println(e);
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        System.out.println("Over");
    }
}
```

OUTPUT:

```
Welcome
Enter valu1
4
Enter value
0
java.lang.ArithmeticException: / by zero
```

b.

```
class ExTy {
    public static void main(String args[])
    {
```

```

try {
    int a[] = { 1, 2, 3, 4, 5 };
    System.out.println(a[5]);
    try {
        int x = a[2] / 0;
    }
    catch (ArithmeticException e2) {
        System.out.println("division by zero is not possible");
    }
}
catch (ArrayIndexOutOfBoundsException e1) {
    System.out.println("ArrayIndexOutOfBoundsException");
    System.out.println("Element at such index does not exists");
}
}
}

```

OUTPUT:

ArrayIndexOutOfBoundsException
Element at such index does not exists

c.

```

class ExTy {
    public static void main(String[] args)
    {
        try {
            System.out.println("inside try block");
            System.out.println(34 / 2);
        }
        catch (ArithmeticException e) {
            System.out.println("Arithmetic Exception");
        }
        finally {
            System.out.println("finally : i execute always.");
        }
    }
}

```

OUTPUT:

inside try block
17
finally : i execute always.

d.

```

class ExTy {
    static void fun()
    {
        try

```

```

        {
            throw new NullPointerException("demo");
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught inside fun().");
            throw e;
        }
    }
}

```

```

public static void main(String args[])
{
    try
    {
        fun();
    }
    catch(NullPointerException e)
    {
        System.out.println("Caught in main.");
    }
}

```

}

OUTPUT:

Caught inside fun().
Caught in main.

e.

```

class ExTy {
    public static void main(String[] args)throws InterruptedException
    {
        Thread.sleep(1000);
        System.out.println("Hello World");
    }
}

```

OUTPUT:

Hello World

f.

```

import java.util.*;
class VotingApplication extends Exception{
    String str;
    VotingApplication(String str){
        this.str = str;
    }
    public String toString(){
        return str;
    }
}
class ExTy {

```



```

static void validatee(int age) throws VotingApplication{
    if(age>=18){
        System.out.println("Welcome to voting");
    }
    else{
        throw new VotingApplication("Not allowed");
    }
}
public static void main(String[] args)throws InterruptedException
{
    int age;
    Scanner in = new Scanner(System.in);
    System.out.println("Enter your age");
    age = in.nextInt();
    try{
        validatee(age);
    }
    catch(VotingApplication e){
        System.out.println("As age is less than 18 "+e);
    }
}
}

```

OUTPUT:

```

Enter your age
17
As age is less than 18 Not allowed

```

g.

```

import java.util.*;
class ExTy {
    public static void main(String[] args)
    {
        int no1, no2, res;
        System.out.println("Welcome");
        Scanner in = new Scanner(System.in);
        try{
            int arr[] = new int[1];
            System.out.println(arr[1]);
            no1 = 20; no2 = 0;
            res = no1/no2;
            System.out.println("Result is "+res);
        }
        catch(Exception ob){
            System.out.println("Generic Exception");
        }
    }
}

```

OUTPUT:

```

Welcome
Generic Exception

```

NAME - PRIYANSHU MALLICK

SIC - 21BCSF11

ROLL. NO - 30, SEC - B, GROUP - B2, BRANCH - CSE