**ABHISHEK SHARMA**

**CS 2ND YEAR**

**SECTION : "I"**

**ROLL NO.: 01**

**ENROLLMENT NO.: 12019009001127**

**Object Oriented Programming using JAVA**

**ASSIGNMENT : 8**

**DATE : 23.04.2021**

**Platform Used : Visual Studio Code and JDK 15**

*Last Date of Submission : 02.05.2021*

**UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**1. Write a Java program to show the use of all keywords for exception handling.**

// Author : Abhishek Sharma, 2021

class q1 {

    public static void main(String[] args) {

        int a=50;

        try {

            int sum=a/0;

        }

        catch(ArithmeticException e){

            System.out.println("Exception: "+ e);

        }

        finally {

            System.out.println("Executing finally block");

        }

    }

}

**Output :**

Exception: java.lang.ArithmeticException: / by zero

Executing finally block

**2. Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception.**

// Author : Abhishek Sharma, 2021

public class q2 {

    public static void main(String[] args) {

        int arr[];

        try {

            arr=new int[-10];

        }

        catch(NegativeArraySizeException f){

            System.out.println("Exception "+f);

        }

        int a=50;

        try{

            int sum=a/0;

        }

        catch(ArithmeticException e){

            System.out.println("Exception: "+e);

        }

    }

}

**Output :**

Exception java.lang.NegativeArraySizeException: -10

Exception: java.lang.ArithmeticException: / by zero

**3. Define an exception called “NoMatchFoundException” that is thrown when a string is not equal to “University”. Write a program that uses this exception.**

// Author : Abhishek Sharma, 2021

class NoMatchFoundException extends Exception{

    NoMatchFoundException(String s){

        super(s);

    }

}

public class q3 {

    public static void Match(String z) throws NoMatchFoundException{

        if(z!="University"){

            throw new NoMatchFoundException("NoMatchFoundException Gen");

        }

    }

    public static void main(String args[]){

        String z="abc";

        try {

            Match(z);

        }

        catch (Exception e) {

            System.out.println(e);

        }

        System.out.print("Rest Code");

    }

}

**Output :**

NoMatchFoundException: NoMatchFoundException Gen

Rest Code

**4. Write a class that keeps a running total of all characters passed to it (one at a time) and throws anexception if it is passed a non-alphabetic character.**

// Author : Abhishek Sharma, 2021

class NonAlphabeticException extends Exception{

    NonAlphabeticException(String s){

        super(s);

    }

}

class Alpha{

    public static void alpha(char a) throws NonAlphabeticException{

        if(a<97 || a>122){

            throw new NonAlphabeticException("NonAlphabeticException");

        }

    }

}

public class q4 {

    public static void main(String args[]){

        char[] a={'a','b','c','1'};

        int i=0;

        while(i<a.length){

            try {

                Alpha.alpha(a[i]);

            }

            catch (NonAlphabeticException e) {

                System.out.println(e);

            }

            i++;

        }

        System.out.println("Rest");

    }

}

**Output :**

NonAlphabeticException: NonAlphabeticException

Rest

**5. Write a program called Factorial.java that computes factorials and catches the result in an array of type long for reuse. The long type of variable has its own range. For example 20! Is as high as the range of long type. So check the argument passes and “throw an exception”, if it is too big or too small.**

** If x is less than 0 throw an IllegalArgumentException with a message “Value of x must be positive”.**

** If x is above the length of the array throw an IllegalArgumentException with a message “Result will overflow”.**

**Here x is the value for which we want to find the factorial.**

// Author : Abhishek Sharma, 2021

public class q5 {

    public static void calculateFactorial(long n) {

        if (n < 0)

            throw new IllegalArgumentException("n must be positive");

        else if (n > 20)

            throw new IllegalArgumentException("n must be < 20");

        else{

            long z=n;

            long fact=1;

            while(n>1){

                fact\*=n;

                n--;

            }

            System.out.println("Factorial of "+z+" = "+fact);

        }

    }

    public static void main(String args[]){

        calculateFactorial(5);

        calculateFactorial(21);

    }

}

**Output :**

Factorial of 5 = 120

Exception in thread "main" java.lang.IllegalArgumentException: n must be < 20

**6. Write a class that keeps a running total of all characters passed to it (one at a time) and throws an exception if it is passed a non-alphabetic character.**

// Author : Abhishek Sharma, 2021

class NonAlphabeticException extends Exception{

    NonAlphabeticException(String s){

        super(s);

    }

}

class Alpha{

    public static void alpha(char a) throws NonAlphabeticException{

        if(a<97 || a>122){

            throw new NonAlphabeticException("NonAlphabeticException");

        }

    }

}

public class q6 {

    public static void main(String args[]){

        char[] a={'a','b','c','1'};

        int i=0;

        while(i<a.length){

            try {

                Alpha.alpha(a[i]);

            }

            catch (NonAlphabeticException e) {

                System.out.println(e);

            }

            i++;

        }

        System.out.println("Rest");

    }

}

**Output :**

NonAlphabeticException: NonAlphabeticException

Rest

**7. Write a program that outputs the name of the capital of the country entered at the command line. The program should throw a “NoMatchFoundException” when it fails to print the capital of the country entered at the command line.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class NoMatchFoundException extends Exception{

    NoMatchFoundException(String s){

        super(s);

    }

}

public class q7 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the capital name");

        String str=sc.next();

        String cap="delhi";

        try {

        if(str.equals(cap)){

            System.out.println("Match");

        }

        else{

            throw new NoMatchFoundException("NoMatchFoundException");

        }

    }

    catch (Exception e) {

        System.out.println(e);

    }

    System.out.print("Rest Code");

    }

}

**Output :**

Enter the capital name

Kolkata

NoMatchFoundException: NoMatchFoundException

Rest Code

**8. Write a program that takes a value at the command line for which factorial is to be computed. The program must convert the string to its integer equivalent. There are three possible user input errors that can prevent the program from executing normally.**

** The first error is when the user provides no argument while executing the program and an ArrayIndexOutOfBoundsException is raised. You must write a catch block for this.**

** The second error is NumberFormatException that is raised in case the user provides a non-integer (float double) value at the command line.**

** The third error is IllegalArgumentException. This needs to be thrown manually if the value at the command line is 0.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

public class q8 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        String str=sc.next();

        int n=0;

        try{

            n=Integer.parseInt(str);

            if(n==0)

                throw new IllegalArgumentException();

            int z=n;

            int fact=1;

            while(n>1){

                fact\*=n;

                n--;

            }

            System.out.println("Factorial of "+z+" = "+fact);

        }

        catch(ArrayIndexOutOfBoundsException e){

            System.out.print(e);

        }

        catch(NumberFormatException e){

            System.out.print(e);

        }

        catch(IllegalArgumentException e){

            System.out.print(e);

        }

    }

}

**Output :**

1.00

java.lang.NumberFormatException: For input string: "1.00"

0

java.lang.IllegalArgumentException

5

Factorial of 5 = 120

**9. Create a user-defined exception named CheckArgument to check the number of arguments passed through the command line. If the number of argument is less than 5, throw the CheckArgumentexception,else print the addition of all the five numbers.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class CheckArgumentexception extends Exception{

    CheckArgumentexception(String s){

        super(s);

    }

}

public class q9 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("enter the limit");

        int n=sc.nextInt();

        System.out.println("enter "+n+" numbers");

        int[] arr=new int[n];

        for(int i=0;i<n;i++)

            arr[i]=sc.nextInt();

        try{

            if(n<5)

                throw new CheckArgumentexception("n is less than 5");

            else{

                int sum=0;

                for(int i=0;i<n;i++)

                    sum+=arr[i];

                System.out.println("Sum is:"+sum);

            }

        }

        catch(Exception e){

            System.out.print(e);

        }

    }

}

**Output :**

enter the limit

4

enter 4 numbers

1

2

3

4

CheckArgumentexception: n is less than 5

**10. Consider a Student examination database system that prints the mark sheet of students. Input the following from the command line.**

**(a) Student’s Name**

**(b) Marks in six subjects**

**These marks should be between 0 to 50. If the marks are not in the specified range, raise a RangeException, else find the total marks and prints the percentage of the students.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class RangeException extends Exception{

    RangeException(){

        super("Marks should be between 0 to 50");

    }

}

public class q10 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the Student name");

        String str=sc.next();

        System.out.println("enter "+6+" marks");

        int[] arr=new int[6];

        for(int i=0;i<6;i++)

            arr[i]=sc.nextInt();

        try{

            int sum=0;

            for(int i=0;i<arr.length;i++){

                if(arr[i]<0 || arr[i]>50)

                    throw new RangeException();

                sum+=arr[i];

            }

            System.out.print("Percentage is : "+(sum/6)\*2);

        }

        catch(RangeException e){

            System.out.print(e);

        }

    }

}

**Output :**

Enter the Student name

Abhishek

enter 6 marks

23

15

10

1000

10

23

RangeException: Marks should be between 0 to 50

Enter the Student name

Abhishek

enter 6 marks

50

50

50

40

40

40

Percentage is : 90

**11. Write a java program to create an custom Exception that would handle at least 2 kind of Arithmetic Exceptions while calculating a given equation (e.g. X+Y\*(P/Q)Z-I)**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class CustomArithmeticException extends Exception{

    CustomArithmeticException(String s){

        super(s);

    }

}

public class q11 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        int x,y,p,q,z,l;

        System.out.println("Enter X");

        x=sc.nextInt();

        System.out.println("Enter Y");

        y=sc.nextInt();

        System.out.println("Enter P");

        p=sc.nextInt();

        System.out.println("Enter P");

        q=sc.nextInt();

        System.out.println("Enter Z");

        z=sc.nextInt();

        System.out.println("Enter L");

        l=sc.nextInt();

        try{

            if(q==0)

                throw new CustomArithmeticException("Cannot divided by 0");

            int sum=(x+y\*(p/q)\*z-l);

            if(sum<0)

                throw new CustomArithmeticException("Sum cannot be negetive");

            System.out.print(sum);

        }

        catch(CustomArithmeticException e){

            System.out.print(e);

        }

    }

}

**Output :**

Enter X

12

Enter Y

10

Enter P

-5

Enter P

0

Enter Z

22

Enter L

10

CustomArithmeticException: Cannot divided by 0

Enter X

2

Enter Y

4

Enter P

6

Enter P

8

Enter Z

10

Enter L

12

CustomArithmeticException: Sum cannot be negetive

**12. Create two user-defined exceptions named “TooHot” and “TooCold” to check the temperature (in Celsius) given by the user passed through the command line is too hot or too cold.**

** If temperature > 35, throw exception “TooHot”.**

** If temperature <5, throw exception “TooCold”.**

** Otherwise, print “Normal” and convert it to Farenheit.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class TooHot extends Exception{

    TooHot(){

        super("Temperature is too-hot");

    }

}

class TooCold extends Exception{

    TooCold(){

        super("Temperature is too-cold");

    }

}

public class q12 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        int x,y,p,q,z,l;

        System.out.println("Enter the temperature (Celsius):");

        x=sc.nextInt();

        try{

            if(x<0)

                throw new TooCold();

            if(x>35)

                throw new TooHot();

            System.out.println("Temperature is (Celsius):"+x);

            System.out.print("Temperature is (Fahrenheit ):"+(x\*(9/5)+32));

        }

        catch(TooHot e){

            System.out.print(e);

        }

        catch(TooCold e){

            System.out.print(e);

        }

    }

}

**Output :**

Enter the temperature (Celsius):

12

Temperature is (Celsius):12

Temperature is (Fahrenheit ):44

Enter the temperature (Celsius):

-20

TooCold: Temperature is too-cold

Enter the temperature (Celsius):

40

TooHot: Temperature is too-hot

**13. Consider an Employee recruitment system that prints the candidate name based on the age criteria. The name and age of the candidate are taken as Input.Create two user-defined exceptions named “TooOlder” and “TooYounger”**

** If age>45, throw exception “TooOlder”.**

** If age<20, throw exception “TooYounger”.**

** Otherwise, print “Eligible” and print the name of the candidate.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class TooYounger extends Exception{

    TooYounger(){

        super("TooYounger");

    }

}

class TooOlder extends Exception{

    TooOlder(){

        super("TooOlder");

    }

}

public class q13 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the Name:");

        String name=sc.next();

        System.out.println("Enter the age:");

        int x=sc.nextInt();

        try{

            if(x<20)

                throw new TooYounger();

            if(x>40)

                throw new TooOlder();

            System.out.println("(Eligible) Name :"+name);

        }

        catch(TooYounger e){

            System.out.print(e);

        }

        catch(TooOlder e){

            System.out.print(e);

        }

    }

}

**Output :**

Enter the Name:

Abhishek

Enter the age:

21

(Eligible) Name :Abhishek

Enter the Name:

Nabarun

Enter the age:

19

TooYounger: TooYounger

Enter the Name:

Sayan

Enter the age:

45

TooOlder: TooOlder

**14. Consider a “Binary to Decimal” Number conversion system which only accepts binary number as Input. If user provides a decimal number a custom Exception “WrongNumberFormat” exception will be thrown. Otherwise, it will convert into decimal and print into the screen.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class WorngNumberException extends Exception{

    WorngNumberException(){

        super("Please enter the binary number");

    }

}

public class q14 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the number (in binary):");

        String str=sc.next();

        try{

            for(int i=0;i<str.length();i++)

                if(str.charAt(i)!='0' && str.charAt(i)!='1')

                    throw new WorngNumberException();

            int a=Integer.parseInt(str,2);

            System.out.print("Decimal :"+a);

        }

        catch(Exception e){

            System.out.print(e);

        }

    }

}

**Output :**

Enter the number (in binary):

111

Decimal :7

Enter the number (in binary):

123

WorngNumberException: Please enter the binary number

**15. Write a Java Program that Implement the Nested Try Statements.**

// Author : Abhishek Sharma, 2021

public class q15 {

    public static void main(String args[]){

        try {

            int a[] = { 1,2,3 };

            System.out.println(a[10]);

            try {

                int x = a[2] / 0;

            }

            catch (ArithmeticException f) {

                System.out.println(f);

            }

        }

        catch (ArrayIndexOutOfBoundsException e) {

            System.out.println(e);

        }

    }

}

**Output :**

java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 3

**16. Write a Java Program to Create Account with 500 Rs Minimum Balance, Deposit Amount, Withdraw Amount and Also Throws LessBalanceException.**

** Java Program Which has a Class Called LessBalanceException Which returns the Statement that Says WithDraw Amount(\_Rs) is Not Valid**

** Java Program that has a Class Which Creates 2 Accounts, Both Account Deposit Money and One Account Tries to WithDraw more Money Which Generates a LessBalanceException Take Appropriate Action for the Same.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class LessBalanceException extends Exception{

    LessBalanceException(int i){

        super("WithDraw Amount "+i+" is not valid");

    }

}

public class q16 {

    public static void main(String[] args) {

        int am=500;

        System.out.println("Account created with initial balance 500");

        int x=0;

        Scanner sc=new Scanner(System.in);

        while(x!=1){

            System.out.println("1. Diposit");

            System.out.println("2. Withdraw");

            int a=sc.nextInt();

            if(a==1){

                System.out.println("Enter the ammount");

                int z=sc.nextInt();

                am+=z;

                System.out.println("Current balance :"+am);

            }

            else if(a==2){

                System.out.println("Enter the ammount");

                int z=sc.nextInt();

                if(z>am)

                    try {

                        throw new LessBalanceException(z);

                    }

                    catch (LessBalanceException e) {

                        System.out.println(e);

                    }

                else

                    am-=z;

                System.out.println("Current balance :"+am);

            }

            else{

                x = 1;

            }

        }

    }

}

**Output :**

Account created with initial balance 500

1. Diposit

2. Withdraw

1

Enter the ammount

1000

Current balance :1500

1. Diposit

2. Withdraw

2

Enter the ammount

200

Current balance :1300

1. Diposit

2. Withdraw

2

Enter the ammount

2000

LessBalanceException: WithDraw Amount 2000 is not valid

Current balance :1300

**17. Consider a Library Management System, where a user wants to find a book. If the book is present in Library (Hint: Use predefined array), then it will print the book. Otherwise it will throw an exception “BookNotFoundException”.**

// Author : Abhishek Sharma, 2021

import java.awt.print.Book;

import java.util.Scanner;

class BookNotFoundException extends Exception{

    BookNotFoundException(){

        super("Book not found exception");

    }

}

public class q17 {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        int books[]={101,202,303,404,505};

        System.out.println("Enter the book no.:");

        int key=sc.nextInt();

        int flag=0;

        for(int i=0;i<books.length;i++){

            if(books[i]==key)

                flag=1;

        }

        try{

            if(flag==0)

                throw new BookNotFoundException();

            else

                System.out.print("Book found");

        }

        catch(BookNotFoundException e){

            System.out.print(e);

        }

    }

}

**Output :**

Enter the book no.:

1111

BookNotFoundException: Book not found exception

Enter the book no.:

101

Book found

**18. Consider a Quiz Management System, where a user needs to answer 5 questions. If any of the answer is wrong, throw an exception “NotCorrectException”. If the answer is correct give a message “good! The answer is correct”.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class NotCorrectException extends Exception{

    NotCorrectException(){

        super("Wrong answer");

    }

}

public class q18 {

    public static void main(String arrg[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Captain of Indian Cricket team?");

        System.out.println("1: Virat Kohli");

        System.out.println("2: MS. Dhoni");

        if(sc.nextInt()==1)

            System.out.println("Good");

        else

        try {

            throw new NotCorrectException();

        }

        catch (NotCorrectException e) {

            System.out.println(e);

        }

        System.out.println("EX Prime Minister of India?");

        System.out.println("1: Dr. Manmohan Singh");

        System.out.println("2: Arvind Kejriwal");

        if(sc.nextInt()==1)

            System.out.println("Good");

        else

        try {

            throw new NotCorrectException();

        }

        catch (NotCorrectException e) {

            System.out.println(e);

        }

        System.out.println("Who wrote the C language?");

        System.out.println("1: Bill Gates");

        System.out.println("2: Dennis Ritchie");

        if(sc.nextInt()==2)

            System.out.println("Good");

        else

        try {

            throw new NotCorrectException();

        }

        catch (NotCorrectException e) {

            System.out.println(e);

        }

        System.out.println("JAVA is written in which year?");

        System.out.println("1: 1992");

        System.out.println("2: 1995");

        if(sc.nextInt()==2)

            System.out.println("Good");

        else

        try {

            throw new NotCorrectException();

        }

        catch (NotCorrectException e) {

            System.out.println(e);

        }

        System.out.println("When was TCS established?");

        System.out.println("1: 1 April 1968");

        System.out.println("2: 1 April 1972");

        if(sc.nextInt()==1)

            System.out.println("Good");

        else

        try {

            throw new NotCorrectException();

        }

        catch (NotCorrectException e) {

            System.out.println(e);

        }

    }

}

**Output :**

Captain of Indian Cricket team?

1: Virat Kohli

2: MS. Dhoni

1

Good

EX Prime Minister of India?

1: Dr. Manmohan Singh

2: Arvind Kejriwal

1

Good

Who wrote the C language?

1: Bill Gates

2: Dennis Ritchie

2

Good

JAVA is written in which year?

1: 1992

2: 1995

1

NotCorrectException: Wrong answer

When was TCS established?

1: 1 April 1968

2: 1 April 1972

1

Good

**19. Write a program to raise a user defined exception if username is less than 6 characters and password does not match.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class PasswordException extends Exception{

    PasswordException(String s){

        super(s);

    }

}

public class q19 {

    public static void main(String arrg[]){

        String pass="UEMKCSEJAVA";

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter the username :");

        String user1=sc.next();

        if(user1.length()<6)

            try {

                throw new PasswordException("Length is less than 6");

            }

            catch (PasswordException e) {

                System.out.print(e);

            }

        System.out.print("Enter the password :");

        String str=sc.next();

        if(str.equals(pass))

            System.out.print("Password Matched");

        else

            try {

                throw new PasswordException("Wrong Password");

            }

            catch (PasswordException e) {

                System.out.print(e);

            }

    }

}

**Output :**

Enter the username :CSE

PasswordException: Length is less than 6Enter the password :CSE00

PasswordException: Wrong Password

Enter the username :ABHISHEK

Enter the password :UEMKCSEJAVA

Password Matched

**20. Write a program to accept a password from the user and throw 'Authentication Failure' exception if the password is incorrect.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class AuthenticationFailure extends Exception{

    AuthenticationFailure(){

        super("AuthenticationFailure");

    }

}

public class q20 {

    public static void main(String arrg[]){

        String pass="UEMKCSEJAVA";

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the password :");

        String str=sc.next();

        if(str.equals(pass))

            System.out.println("Password Matched");

        else

            try {

                throw new AuthenticationFailure();

            }

            catch (AuthenticationFailure e) {

                System.out.print(e);

            }

    }

}

**Output :**

Enter the password :

UEMKCSEJAVA

Password Matched

Enter the password :

KOLKATAJAVA

AuthenticationFailure: AuthenticationFailure

**21. Write a program to input name and age of a person and throw a user-defined exception, if the entered age is negative.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class NegAgeException extends Exception{

    NegAgeException(){

        super("Age should'nt be negetive");

    }

}

public class q21 {

    public static void main(String arrg[]){

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter the name :");

        String user1=sc.next();

        System.out.println("Enter the age :");

        int age=sc.nextInt();

        if(age<0)

            try {

                throw new NegAgeException();

            }

            catch (NegAgeException e) {

                System.out.print(e);

            }

    }

}

**Output :**

Enter the name :Abhishek

Enter the age :

-21

NegAgeException: Age should'nt be negetive

Enter the name :Sayan

Enter the age :

21

**22. Write a program to throw user defined exception if the given number is not positive.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class NotPositiveException extends Exception{

    NotPositiveException(){

        super("Age should'nt be negetive");

    }

}

public class q22{

    public static void main(String arrg[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter a number :");

        int age=sc.nextInt();

        if(age<0)

            try {

                throw new NotPositiveException();

            }

            catch (NotPositiveException e) {

                System.out.print(e);

            }

    }

}

**Output :**

Enter a number :

-65

**23. Write a program to throw a user-defined exception "String Mismatch Exception", if two strings are not equal (ignore the case).**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class StringMismatchException extends Exception{

    StringMismatchException (){

        super("StringMismatchException ");

    }

}

public class q23 {

    public static void main(String arrg[]){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter 1st String :");

        String str=sc.next();

        System.out.println("Enter 2nd String :");

        String str1=sc.next();

        if(str.equals(str1))

            System.out.println("String Matched");

        else

        try {

            throw new StringMismatchException();

        }

        catch (StringMismatchException e) {

            System.out.print(e);

        }

    }

}

**Output :**

Enter 1st String :

Abhishek

Enter 2nd String :

Abhishek

String Matched

Enter 1st String :

Abhishek

Enter 2nd String :

Sharma

StringMismatchException: StringMismatchException

**24. Design a stack class. Provide your own stack exceptions namely push exception and pop exception, which throw exceptions when the stack is full and when the stack is empty respectively. Show the usage of these exceptions in handling a stack object in the main.**

// Author : Abhishek Sharma, 2021

class StackException extends Exception{

    public StackException(){

        super("Stack size exception");

    }

}

class Stack\_{

    static int arr[]=new int[5];

    static int max=arr.length;

    static int top=-1;

    static void push(int a){

        top++;

        if(top<max){

            arr[top]=a;

            System.out.println(a+" pushed");

        }

        else{

            top--;

            try {

                throw new StackException();

            }

            catch (StackException e) {

                System.out.println(e);

            }

        }

    }

    static void pop(){

        if(top>=0)

            System.out.println(arr[top--]+" Popped");

        else{

            top++;

            try {

                throw new StackException();

            }

            catch (StackException e) {

                System.out.println(e);

            }

        }

    }

}

public class q24 {

    public static void main(String[] args) {

        Stack\_.push(10);

        Stack\_.push(20);

        Stack\_.push(30);

        Stack\_.push(40);

        Stack\_.push(50);

        Stack\_.push(50);

        Stack\_.pop();

        Stack\_.pop();

        Stack\_.pop();

        Stack\_.pop();

        Stack\_.pop();

        Stack\_.pop();

    }

}

**Output :**

10 pushed

20 pushed

30 pushed

40 pushed

50 pushed

StackException: Stack size exception

50 Popped

40 Popped

30 Popped

20 Popped

10 Popped

StackException: Stack size exception

**25. Write an application that displays a series of at least five student ID numbers (that you have stored in an array) and asks the user to enter a numeric test score for the student. Create a ScoreException class, and throw a ScoreException for the class if the user does not enter a valid score (zero to 100). Catch the ScoreException and then display an appropriate message. In addition, store a 0 for the student’s score. At the end of the application, display all the student IDs and scores.**

// Author : Abhishek Sharma, 2021

import java.util.Scanner;

class ScoreException extends Exception{

    ScoreException (){

        super("Invaid Score");

    }

}

public class q25{

    public static void main(String arrg[]){

        int enroll[]={101,102,103,104,105};

        int marks[]=new int[5];

        Scanner sc=new Scanner(System.in);

        for(int i=0;i<enroll.length;i++){

            System.out.println("Enter marks of Enrollment no:"+enroll[i]);

            int a=sc.nextInt();

            if(a<0 || a>100){

                try {

                    throw new ScoreException();

                }

                catch (ScoreException e) {

                    System.out.println(e);

                }

                marks[i]=0;

            }

            else{

                marks[i]=a;

            }

        }

        for(int i=0;i<enroll.length;i++){

            System.out.println("Enrollment no:"+enroll[i]+" Marks:"+marks[i]);

        }

    }

}

**Output :**

Enter marks of Enrollment no:101

98

Enter marks of Enrollment no:102

56

Enter marks of Enrollment no:103

-45

ScoreException: Invaid Score

Enter marks of Enrollment no:104

80

Enter marks of Enrollment no:105

-65

ScoreException: Invaid Score

Enrollment no:101 Marks:98

Enrollment no:102 Marks:56

Enrollment no:103 Marks:0

Enrollment no:104 Marks:80

Enrollment no:105 Marks:0

**-- O --**