**Practical No.10**

**Name**: Anuj Rajendra Mane

**Div**: SY A

**Roll No**: 65 **Batch**: A3

**PRN** **No**: 2162701242054

**Title: Program on Exception Handling**

**Theory:**

**Exception Handling:**

To understand the exception firstly we should know about the error in prgram.Errors in program can be categorized into two types.

**Compile Time Errors**

**Run Time Errors**

Compile Time Errors:- Errors caught during compiled time is called Compile time errors. Compile time errors include library reference, syntax error or incorrect class import.

Run Time Errors:- The error that occours during the run time of program is called run time error.

They are also known as exceptions. Hence we can say that exception is a runtime error that occours because of user's mistake.

**Reasons of Exception**

Mismatched Input :Suppose that we are entering our name in place of age,causing exception because age is of data type int and name will be string.

File does not exist :Suppose that we are reading a text file easy.txt and that file does not exist in the system,causing exception.

Exception related to array :Suppose that the array size is 5 and we are inserting more than 5 elements,causing exception.

Divide by zero exception :When a number is divided by zero then the output will be undefined(infinity).

**Exception Handling**

The process of handling the exception is called exception handling.There are four main keyword are used to solve the problem of exception.

try block :It is the place where actual code is written and exception occours.when the code will lead to any error, that error/exception will get caught inside the catch block.

catch block : catch block is intended to catch the error and handle the exception condition. We can have multiple catch blocks to handle different types of exception and perform different actions when the exceptions occur.

throw statement : It is used to show the user defined message about the exception.

finally block : This block executes either exception occours or does not occours.

**Syntax of try-catch**

try {

// protected code

} catch( ExceptionName e1 ) {

// catch block

} catch( ExceptionName e2 ) {

// catch block

} catch( ExceptionName eN ) {

// catch block

}

**Example 1:try \_ catch**

import java.util.Scanner;

class Easy

{

public static void main(String[] args)

{

Scanner obj=new Scanner(System.in);

int a,b,c;

try

{

System.out.println("Enter first number");

b=obj.nextInt();

System.out.println("Enter second number");

c=obj.nextInt();

a=b/c;

System.out.println("Div="+a);

} catch (Exception e)

{

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

}

}

}

/\*

### Output ###

First Run:

Enter first number

45

Enter second number

9

Div=5

Second Run:

Enter first number

45

Enter second number

0

Error:java.lang.ArithmeticException: / by zero

\*/

**Example 2:try \_ catch \_ throw**

import java.util.Scanner;

class Easy

{

public static void main(String[] args)

{

Scanner obj=new Scanner(System.in);

int a,b,c;

try

{

System.out.println("Enter first number");

b=obj.nextInt();

System.out.println("Enter second number");

c=obj.nextInt();

if(c!=0)

{

a=b/c;

System.out.println("Div="+a);

}

else

throw new Exception("Don't put zero in denominator");

} catch (Exception e)

{

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

}

}

}

/\*

### Output ###

First Run:

Enter first number

45

Enter second number

9

Div=5

Second Run:

Enter first number

45

Enter second number

0

Error:java.lang.Exception: Don't put zero in denominator

\*/

**Example 3:try \_ catch \_ throw \_ finally**

import java.util.Scanner;

class Easy

{

public static void main(String[] args)

{

Scanner obj=new Scanner(System.in);

int a=0,b,c;

try

{

System.out.println("Enter first number");

b=obj.nextInt();

System.out.println("Enter second number");

c=obj.nextInt();

if(c!=0)

{

a=b/c;

}

else

throw new Exception("Don't put zero in denominator");

} catch (Exception e)

{

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

}

finally

{

System.out.println("Div="+a);

}

}

}

/\*

### Output ###

First Run:

Enter first number

45

Enter second number

9

Div=5

Second Run:

Enter first number

45

Enter second number

0

Error:java.lang.Exception: Don't put zero in denominator

Div=0

\*/

**Conclusion:**

I can learned the concepts of eception handling and also doing the programs of try-catch-throw-finally programs.

**Completion Date: Co-Ordinator Sign:**