WEEK-5

(A) AIM

To write a java program to demonstrate exception handling mechanism.

THEORY

JAVA TRY BLOCK

Java try block is used to enclose the code that might throw an exception. It must be used within the method.

JAVA CATCH BLOCK

Java catch block is used to handle the Exception by declaring the type of exception within the parameter. The declared exception must be the parent class exception (i.e., Exception) or the generated exception type. The catch block must be used after the try block only. You can use multiple catch block with a single try block.

ALGORITHM

STEP 1 : START
STEP 2 : Create a class with name 'MultipleCatchBlocks' and write main() STEP 3 : Start try block
STEP 4 : Read two parameters
STEP 5 : Evaluate Division of two parameters
STEP 6 : End try block
STEP 7 : Write catch block to handle ArithmeticException
STEP 8 : Write catch block to handle ArrayIndexOutOfBoundsException
STEP 9 : Write catch block to handle NumberFormatException
STEP 10 : Write Finally block
STEP 11 : Print message 'Exception Handling completed..'
STEP 12 : STOP

SOURCE CODE

```
public class MultipleCatchBlock1 {

   public static void main(String[] args) {

       try{
          int a[]=new int[5];
          a[6]=30/0;
         }
        catch(ArithmeticException e)
          {
           System.out.println("Arithmetic Exception occurs");
          }
        catch(ArrayIndexOutOfBoundsException e)
          {
```

```
            System.out.println("ArrayIndexOutOfBounds Exception occurs");
          }
        catch(Exception e)
          {
           System.out.println("Parent Exception occurs");
          }
        System.out.println("rest of the code");
    }
}
```

OUTPUT

```
Arithmetic Exception occurs
rest of the code
```

(B) AIM: To write a java program to create and test user defined exception class.

THEORY: Creating our own Exception known as custom exception or user-defined exception. Java custom exceptions are used to customize the exception according to user need.By the help of custom exception, we can have your own exception and message.The Exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.

Java try block

Java try block is used to enclose the code that might throw an exception. It must be used within the method.

Java catch block

Java catch block is used to handle the Exception by declaring the type of exception within the parameter. The declared exception must be the parent class exception (i.e., Exception) or the generated exception type. The catch block must be used after the try block only. You can use multiple catch block with a single try block.

ALGORITHM :

STEP1:START
STEP2: Create a class with name ‘InvalidAgeException’ extends exception
STEP3: Define a parameterized constructor InvalidAgeException(string s)
STEP4. Call parent constructor in the derived class constructor using super(s) keyword.
STEP5: Close the custom exception class ‘InvalidAgeException’
STEP6: Create a class with name ‘TestCustomException1’
STEP7:Define static method validate(int age) throws custom exception ‘InvalidAgeException’
STEP8: Validate age: if age<18 then throw new InvalidAgeException("not valid")
STEP9: else print ("welcome to vote")
STEP10: Define main()
STEP11: call validate() static method with one parameter value in the try and catch block . Trt and catch block will throws an exception if any
STEP12: print message ‘Custom exception Demo completed..’
STEP13: STOP

SOURCE CODE:

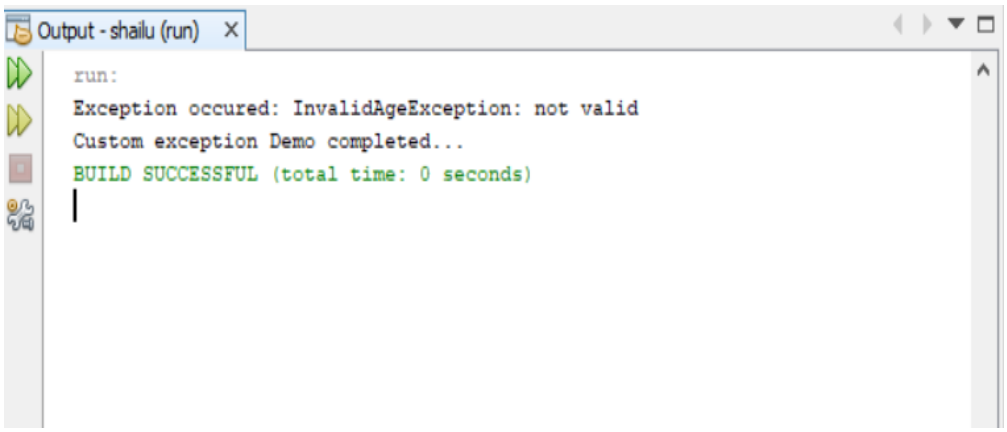
```
class InvalidAgeException extends Exception{
          InvalidAgeException(String s){
           super(s);
          }
         }
         class TestCustomException1{

           static void validate(int age)throws InvalidAgeException{
             if(age<18)
              throw new InvalidAgeException("not valid");
             else
              System.out.println("welcome to vote");
           }

           public static void main(String args[]){
             try{
             validate(13);
             }catch(Exception m){System.out.println("Exception occured: "+m);}

             System.out.println("Custom exception Demo completed...");
          }
         }
```

OUTPUT



VIVA - VOCE:

1. What is an Exception?

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

2. What is the importance of Exception handling in java?

The core advantage of **exception handling** is to maintain the normal flow of the application. An **exception** normally disrupts the normal flow of the application that is why we use **exception handling**.

3. Define Try and catch Blocks?

Java try block is used to enclose the code that might throw an exception. The try **block** contains set of statements where an exception can occur. A try **block** is always followed by a**catch block**, which handles the exception that occurs in associated try **block**. A try **block** must be followed by**catch blocks** or finally **block** or both.

4. What is the purpose of 'finally' block?

The finally block in java is used to put important codes such as clean up code e.g. closing the file or closing the connection. The finally block executes whether exception rise or not and whether exception handled or not. A finally contains all the crucial statements regardless of the exception occurs or not.

5. What is the Difference between Throw and throws?

Throw	**Throws**
Throw keyword is used **in the** method body to **throw** an exception, while **throws** is used in method signature to declare the exceptions that can occur **in the** statements present **in the** method.	**Throws** clause is used to declare an exception, which means it works similar to the try-catch block.