Ex. 7 **EXCEPTION HANDLING**

Date: 30-08-2024

PROGRAM 1

AIM:

To verify whether the given array size is negative and handle the exception using a try-catch block.

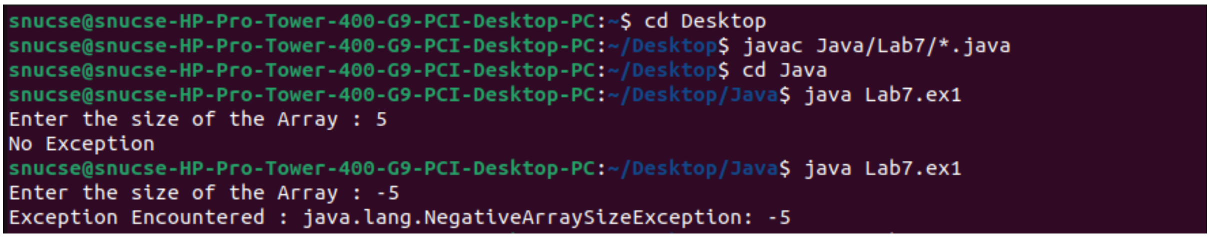
ALGORITHM:

1. Receive the array size as input from the user.
2. Use a try block to attempt to create an array with the given size.
3. If the size is negative, an exception is thrown.
4. Catch the exception using a catch block and display the error message.

PROGRAM:

```
package Lab7;

import java.util.*;

public class ex1 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the size of the Array : ");
        int length = input.nextInt();
        try {
            int arr[] = new int[length];
            System.out.println("No Exception");
        } catch (NegativeArraySizeException e) {
            System.out.print("Exception Encountered : ");
            System.out.println(e);
        }
    }
}
```

OUTPUT:



```
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ cd Desktop
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop$ javac Java/Lab7/*.java
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop$ cd Java
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex1
Enter the size of the Array : 5
No Exception
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex1
Enter the size of the Array : -5
Exception Encountered : java.lang.NegativeArraySizeException: -5
```

PROGRAM 2

AIM:

To create a Java program to check if the input age is valid for voting, throwing an exception if the age is less than 18.

ALGORITHM:

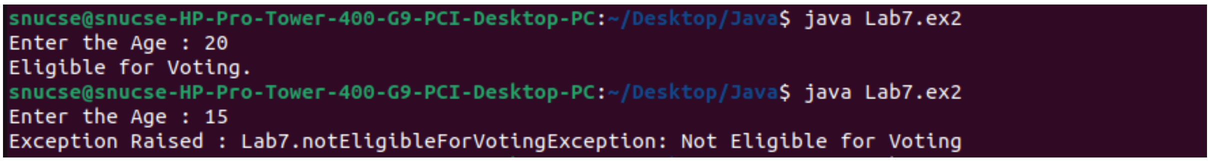
1. Create a custom exception class that extends the Exception class. Inside the method, check if the age is less than 18.
2. If the age is less than 18, use the throw an exception and indicate that the person is not eligible to vote.
3. In the calling code, handle the exception using a try-catch block and display an appropriate message.
4. If the age is 18 or above, print a message indicating eligibility to vote.

PROGRAM:

```
package Lab7;

import java.util.Scanner;

class notEligibleForVotingException extends Exception {
   public notEligibleForVotingException(String content) {
      super(content);
   }
}
```

```
public class ex2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the Age : ");
        int age = input.nextInt();
        try {
            if (age < 18) {
                throw new notEligibleForVotingException("Not Eligible for
Voting");
            } else {
                System.out.println("Eligible for Voting.");
            }
        } catch (notEligibleForVotingException e) {
            System.out.println("Exception Raised : " + e);
        }
    }
}
```

OUTPUT:



```
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex2
Enter the Age : 20
Eligible for Voting.
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex2
Enter the Age : 15
Exception Raised : Lab7.notEligibleForVotingException: Not Eligible for Voting
```

PROGRAM 3

AIM:

To create a Java program that demonstrates handling multiple exceptions in a single program using multiple catch blocks.

ALGORITHM:

1. Create and initialise multiple exceptions like ArrayIndexOutOfBounds, Division by 0 and IllegalArgument in a single program.
2. Use a menu driven program to prompt the user to enter the exception that needs to be executed and displayed.
3. Use multiple catch statements to display the Exception.
4. Display the error message.

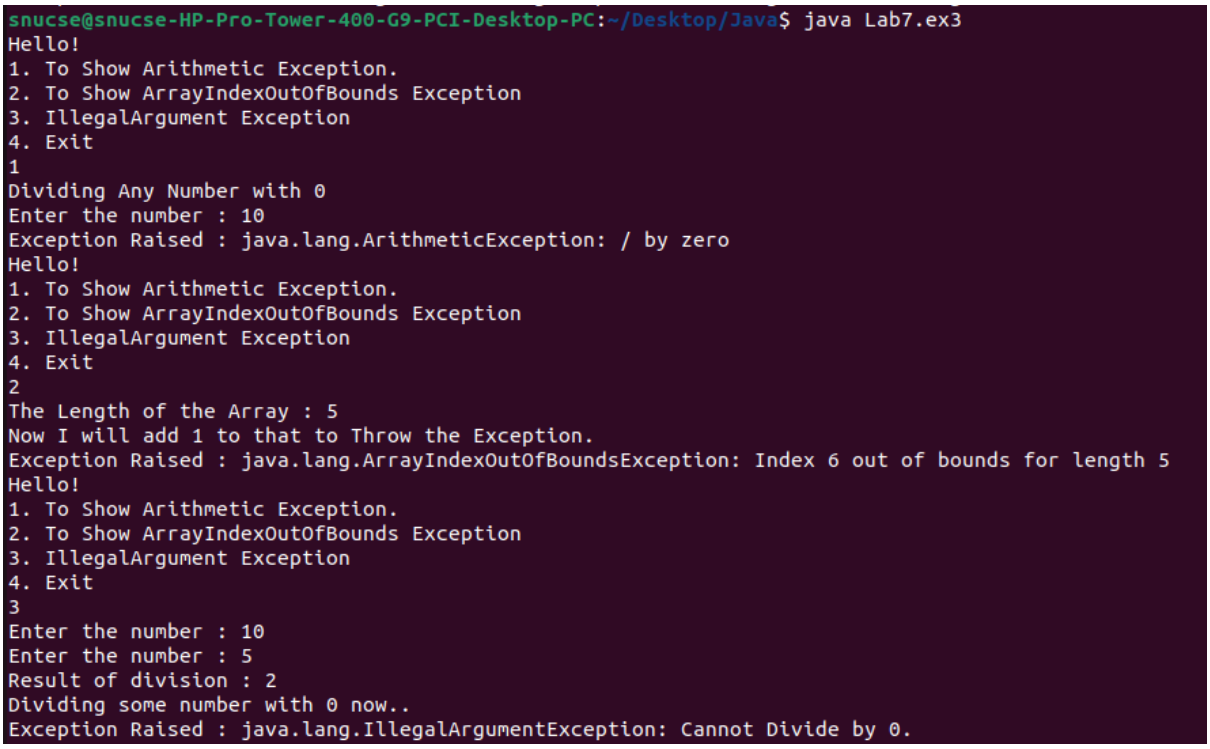
PROGRAM:

```
package Lab7;

import java.util.*;

public class ex3 {
   int divide(int a) {
      return a / 0;
   }

   int divide(int a, int b) {
      if (b > 0)
          return a / b;
      else
          throw new IllegalArgumentException("Cannot Divide by 0.");
   }

   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      ex3 obj = new ex3();
      int choice;
      int num1;
      int num2;
      boolean loopController = true;
      while (loopController) {
          System.out.println("Hello!");
          System.out.println("1. To Show Arithmetic Exception.");
                System.out.println("2.  To  Show  ArrayIndexOutOfBounds
Exception");
          System.out.println("3. IllegalArgument Exception");
          System.out.println("4. Exit");
          try {
              choice = input.nextInt();
              switch (choice) {
                  case 1:
                      System.out.println("Dividing Any Number with 0");
```

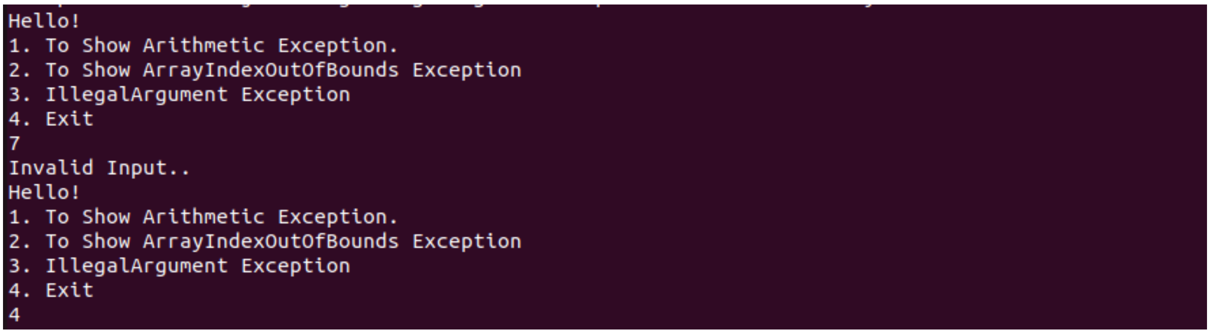
```
          System.out.print("Enter the number : ");
          num1 = input.nextInt();
          System.out.println(obj.divide(num1));
          break;
        case 2:
          int arr[] = { 1, 2, 3, 4, 5 };
              System.out.println("The Length of the Array : " +
arr.length);
           System.out.println("Now I will add 1 to that to Throw
the Exception.");
          System.out.println(arr[arr.length + 1]);
          break;
        case 3:
          System.out.print("Enter the number : ");
          num1 = input.nextInt();
          System.out.print("Enter the number : ");
          num2 = input.nextInt();
                System.out.println("Result of division : " +
obj.divide(num1, num2));
              System.out.println("Dividing some number with 0
now..");
          System.out.println(obj.divide(num1, 0));
          break;
        case 4:
          loopController = false;
```

```
                        break;
                    default:
                        System.out.println("Invalid Input..");
                        break;
                }
                                    } catch (ArrayIndexOutOfBoundsException
arrayIndexOutOfBoundsException) {
                                    System.out.println("Exception Raised : " +
arrayIndexOutOfBoundsException);
            } catch (ArithmeticException arithmeticException) {
                                    System.out.println("Exception Raised : " +
arithmeticException);
            } catch (IllegalArgumentException illegalArgumentException)
{
                                    System.out.println("Exception Raised : " +
illegalArgumentException);
            } catch (Exception e) {
                System.out.println("Exception Raised : " + e);
            }
        }
    }
}
```

OUTPUT:



```
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex3
Hello!
1. To Show Arithmetic Exception.
2. To Show ArrayIndexOutOfBounds Exception
3. IllegalArgument Exception
4. Exit
1
Dividing Any Number with 0
Enter the number : 10
Exception Raised : java.lang.ArithmeticException: / by zero
Hello!
1. To Show Arithmetic Exception.
2. To Show ArrayIndexOutOfBounds Exception
3. IllegalArgument Exception
4. Exit
2
The Length of the Array : 5
Now I will add 1 to that to Throw the Exception.
Exception Raised : java.lang.ArrayIndexOutOfBoundsException: Index 6 out of bounds for length 5
Hello!
1. To Show Arithmetic Exception.
2. To Show ArrayIndexOutOfBounds Exception
3. IllegalArgument Exception
4. Exit
3
Enter the number : 10
Enter the number : 5
Result of division : 2
Dividing some number with 0 now..
Exception Raised : java.lang.IllegalArgumentException: Cannot Divide by 0.
```



```
Hello!
1. To Show Arithmetic Exception.
2. To Show ArrayIndexOutOfBounds Exception
3. IllegalArgument Exception
4. Exit
7
Invalid Input..
Hello!
1. To Show Arithmetic Exception.
2. To Show ArrayIndexOutOfBounds Exception
3. IllegalArgument Exception
4. Exit
4
```

PROGRAM 4

AIM:

To create a Java program to demonstrate exception handling using the throw keyword and ensure a block of code executes using the finally keyword.

ALGORITHM:

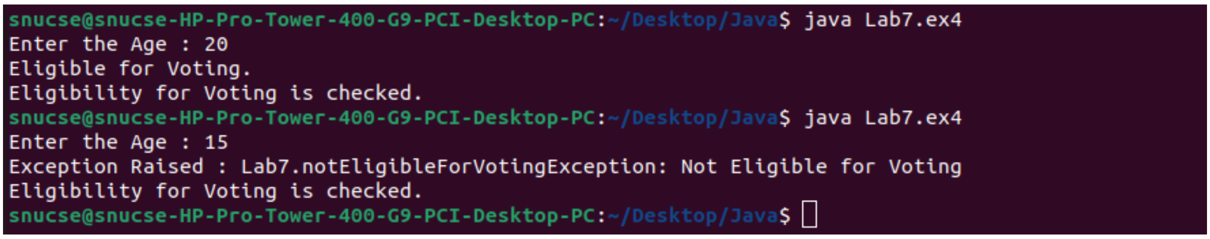
1. Create a program to check the Eligibility to Vote using Exception Handling.
2. In the main code, use a try-catch block to handle the exception.
3. Use a finally block after the try-catch block to ensure that the program has been executed.
4. Display appropriate messages indicating the exception and the execution of the finally block.

PROGRAM:

```
package Lab7;

import java.util.*;

public class ex4 {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.print("Enter the Age : ");
      int age = input.nextInt();
```

```
        try {
            if (age < 18) {
                throw new notEligibleForVotingException("Not Eligible for Voting");
            } else {
                System.out.println("Eligible for Voting.");
            }
        } catch (notEligibleForVotingException e) {
            System.out.println("Exception Raised : " + e);
        } finally {
            System.out.println("Eligibility for Voting is checked.");
        }
    }
}
```

OUTPUT:



```
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex4
Enter the Age : 20
Eligible for Voting.
Eligibility for Voting is checked.
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ java Lab7.ex4
Enter the Age : 15
Exception Raised : Lab7.notEligibleForVotingException: Not Eligible for Voting
Eligibility for Voting is checked.
snucse@snucse-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Desktop/Java$ 
```

RESULT:

Thus, different Java Applications to execute Exception Handling have been compiled and executed successfully.