

## Lab-9-WT lab

By: - Anusthan Singh (20051337)

Q1. Write a Java program to generate an `ArrayIndexOutOfBoundsException` and handle it using catch statement.

Code:-

```
import java.util.Arrays;
import java.util.Scanner;

public class q1_AIOBE {
    public static void main(String args[]) {
        int[] Array = {234, 43, 65, 70, 68, 34, 33};
        System.out.println("Elements in the array are: ");
        System.out.println(Arrays.toString(Array));
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the index of the required element :");
        try {
            int element = sc.nextInt();
            System.out.println("Element in the given index is : "+Array[element]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("The index you have entered is invalid, Not possible");
            System.out.println("Please enter an index number between 0 and 6");
        }
    }
}
```

Output:-

Elements in the array are:

[234, 43, 65, 70, 68, 34, 33]

Enter the index of the required element :

4

Element in the given index is : 68

Or

Elements in the array are:

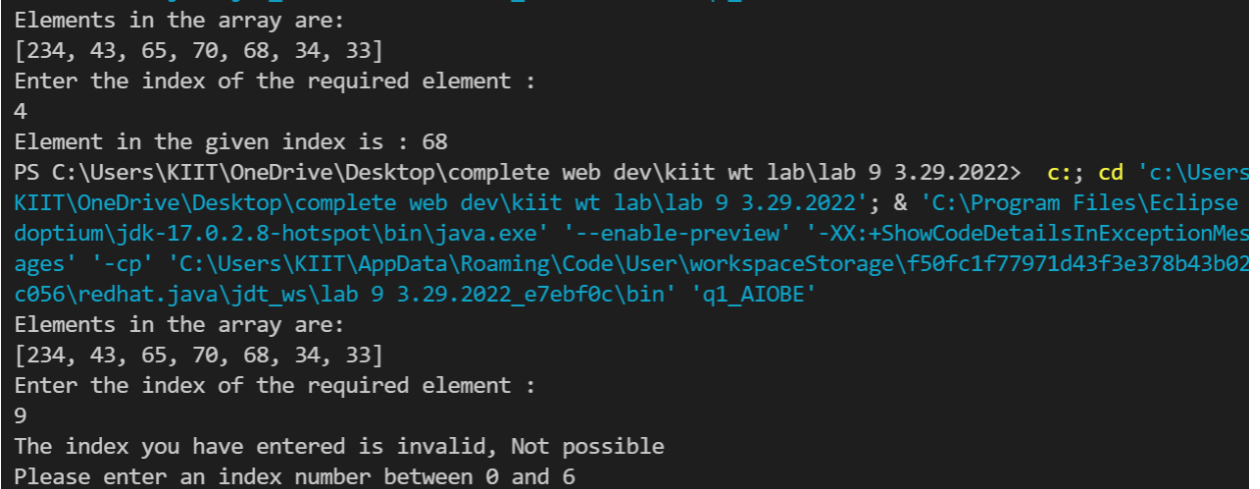
[234, 43, 65, 70, 68, 34, 33]

Enter the index of the required element : 9

The index you have entered is invalid, Not possible

Please enter an index number between 0 and 6

Screenshoot:-



```
Elements in the array are:
[234, 43, 65, 70, 68, 34, 33]
Enter the index of the required element :
4
Element in the given index is : 68
PS C:\Users\KIIT\OneDrive\Desktop\complete web dev\kiit wt lab\lab 9 3.29.2022> cd 'c:\Users\KIIT\OneDrive\Desktop\complete web dev\kiit wt lab\lab 9 3.29.2022'; & 'C:\Program Files\Eclipse\
doptium\jdk-17.0.2.8-hotspot\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMes
ages' '-cp' 'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\f50fc1f77971d43f3e378b43b02
c056\redhat.java\jdt_ws\lab 9 3.29.2022_e7ebf0c\bin' 'q1_AI0BE'
Elements in the array are:
[234, 43, 65, 70, 68, 34, 33]
Enter the index of the required element :
9
The index you have entered is invalid, Not possible
Please enter an index number between 0 and 6
```

Q2. Write a Java program to illustrate try..catch..finally block.

```
class q2
{
    public static void main(String[] args)
    {
        int i = 80, j = 4, k = 4;
        int answer;
        try
        {
            answer = i / (j-k);
        }
        catch (ArithmeticException ae)
        {
            System.out.println("Cannot divided by zero."+ae);
        }
    }
}
```

```

    }
    finally
    {
        System.out.println("finally block");
    }
    answer = i / (j+k);
    System.out.println("Answer: "+answer);
}
}

```

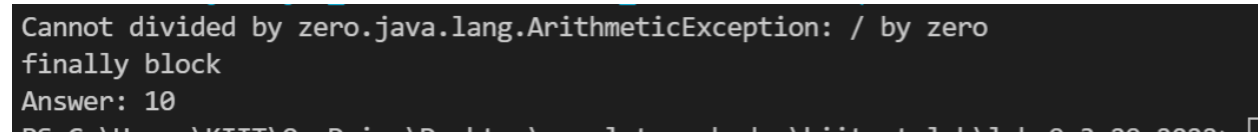
Output:-

Cannot divided by zero.java.lang.ArithmeticException: / by zero

finally block

Answer: 10

Screenshot:-



```

Cannot divided by zero.java.lang.ArithmeticException: / by zero
finally block
Answer: 10

```

Q3 Write a Java class which has a method called ProcessInput().

This method checks the number entered by the user.

If the entered number is negative then throw an user defined exception called NegativeNumberException, otherwise it displays the double value of the entered number.

Code:-

```

import java.util.*;
class q3
{
    public static void ProcessInput(int num) throws NegativeNumberException
    {
        if(num<0)
        {
            throw new NegativeNumberException(num);
        }
        else
    }
}

```

```

        {
            System.out.println("The Double of the entered number: "+(num+num));
        }
    }
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        int num;
        System.out.println("Enter a number:");
        num = input.nextInt();
        try
        {
            ProcessInput(num);
        }
        catch (NegativeNumberException e)
        {
            System.out.println(e);
        }
        input.close();
    }
}
class NegativeNumberException extends Exception
{
    int e;
    NegativeNumberException(int num)
    {
        e = num;
    }
}

```

Output:-

Enter a number:

5

The Double of the entered number: 10

**Or**

Enter a number:

-5

NegativeNumberException

Screenshot:-

```
Enter a number:
5
The Double of the entered number: 10
PS C:\Users\KIIT\OneDrive\Desktop\complete web dev\kiit wt lab\lab 9 3.29.2022> c::; cd 'c:\Users\
KIIT\OneDrive\Desktop\complete web dev\kiit wt lab\lab 9 3.29.2022'; & 'C:\Program Files\Eclipse A
doptium\jdk-17.0.2.8-hotspot\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMess
ages' '-cp' 'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\f50fc1f77971d43f3e378b43b02f
c056\redhat.java\jdt_ws\lab 9 3.29.2022_e7ebf0c\bin' 'q3'
Enter a number:
-5
NegativeNumberException
```

Q4. Write a program to create user defined exceptions called HrsException, MinException and SecException.

Create a class Time which contains data members hours, minutes, seconds and throw the user defined exceptions

if hours (>24 & <0), minutes(>60 & <0), seconds(>60 & <0).

Code:-

Enter hours, minutes and seconds

08 30 25

Time is 8:30:25

Screenshot:-

```
Enter hours, minutes and seconds
08 30 25
Time is 8:30:25
```

Q5. Write a java program to implement a stack class having methods push () and pop().

These methods must be designed to throw user defined exception when the stack is empty or full.

Code:-

```
class q5_Stack {

    //to store elements of stack
    private int arr[];

    private int top;

    private int capacity;

    // Creating a stack
    q5_Stack(int size) {

        arr = new int[size];
        capacity = size;
        top = -1;
    }

    public void push(int x) {
        if (isFull()) {
            System.out.println("Stack OverFlow");

            System.exit(1);
        }

        System.out.println("Insertion = " + x);
        arr[++top] = x;
    }

    public int pop() {

        if (isEmpty()) {
            System.out.println("STACK EMPTY");

            System.exit(1);
        }
    }
}
```

```

        return arr[top--];
    }

    public int getSize() {
        return top + 1;
    }

    public Boolean isEmpty() {
        return top == -1;
    }

    public Boolean isFull() {
        return top == capacity - 1;
    }

    public void printStack() {
        for (int i = 0; i <= top; i++) {
            System.out.print(arr[i] + ", ");
        }
    }

    public static void main(String[] args) {
        q5_Stack stack = new q5_Stack(5);

        stack.push(10);
        stack.push(15);
        stack.push(20);

        System.out.print("Stack: ");
        stack.printStack();

        stack.pop();
        System.out.println("\n After popping out");
        stack.printStack();

    }
}

```

Output:-

Insertion 10

Insertion 15

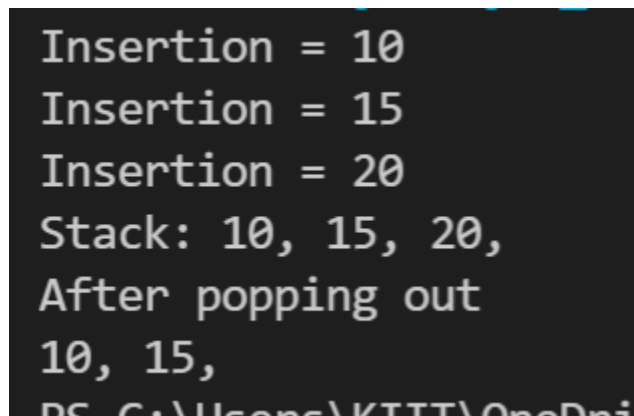
Insertion 20

Stack: 10, 15, 20,

After popping out

10, 15,

Screenshot:-

A screenshot of a terminal window with a black background and white text. The text shows a sequence of stack operations: three insertion steps (10, 15, 20), followed by the current stack state (10, 15, 20), and then two popping steps (10, 15). The path 'PS C:\Users\KITT\OneDri' is partially visible at the bottom.

```
Insertion = 10
Insertion = 15
Insertion = 20
Stack: 10, 15, 20,
After popping out
10, 15,
PS C:\Users\KITT\OneDri
```

**By:- Anusthan Singh (20051337)**