Day 9 -Aditya - 13th June 2025

| Inheritance, Polymorphism, Encapsulation, Abstraction, Interfaces, Exception Handling, |
| --- |
| Collections Framework intro, Streams, File I/O, Multithreading overview |

Exceptions:

class -----> class ======> extends ====> no multiple inheritance

interface ----> class =====> Implements

interface ---> interface ====> extends

Exceptions: --> unexpected situations

2 categoried

- Checked ---> compile time

- Unchecked ---> Runtime time..

pre defined exception classes

FileNotFound

ArrayIndexoutofBounds exception

NullPointer Exception

Arthematic Exception ...

user defined Exceptions

try catch blocks

try{

}catch(Exception ex){

}catch(ArthematicException ex1{

}

finally{

sout(" ");

}

Task 1:

public class Main {

public static void main(String[] args) {

int[] myNumbers = {1, 2, 3};

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

}

}

public class Main {

public static void main(String[] args) {

int[] myNumbers = {1, 2, 3};

System.*out*.println(myNumbers[10]);

}

}

Output:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 3

at Main.main(Main.java:4)

Process finished with exit code 1

Task 2:

public class Main {

public static void main(String[] args) {

try {

int[] myNumbers = {1, 2, 3};

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

} catch (Exception e) {

System.out.println("Something went wrong.");

}

}

}

public class Practice {

public static void main(String[] args) {

try {

int[] myNumbers = {1, 2, 3};

System.*out*.println(myNumbers[10]);

} catch (Exception e) {

System.*out*.println("Something went wrong.");

}

}

}

Output:

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.2\lib\idea\_rt.jar=54756" -Dfile.encoding=UTF-8 -classpath C:\Users\adishum\IdeaProjects\ATLAS\_DAY\_9\out\production\ATLAS\_DAY\_9 Practice

Picked up JAVA\_TOOL\_OPTIONS: -Dlog4j2.formatMsgNoLookups=true

Something went wrong.

Process finished with exit code 0

Task 3:

public class Main {

public static void main(String[] args) {

try {

int[] myNumbers = {1, 2, 3};

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

} catch (Exception e) {

System.out.println("Something went wrong.");

} finally {

System.out.println("I'm from finally block.");

}

}

}

public class Main {

public static void main(String[] args) {

try {

int[] myNumbers = {1, 2, 3};

System.*out*.println(myNumbers[10]);

} catch (Exception e) {

System.*out*.println("Something went wrong.");

} finally {

System.*out*.println("I'm from finally block.");

}

}

}

Output:

Something went wrong.

I'm from finally block.

Process finished with exit code 0

Task 4:

from the above code from task 3 .. try accessing the 2 or the 3rd index and check the out..

By 10.23 plz finish till task 4..

public class Main {

public static void main(String[] args) {

int[] myNumbers = {1, 2, 3}; // Valid indexes are 0, 1, and 2

// Accessing index 2 (3rd element)

try {

System.*out*.println("Accessing index 2:");

System.*out*.println(myNumbers[2]); // This will work (value is 3)

} catch (Exception e) {

System.*out*.println("Something went wrong.");

} finally {

System.*out*.println("I'm from finally block.");

}

// Accessing index 3 (4th element - doesn't exist)

try {

System.*out*.println("\nAccessing index 3:");

System.*out*.println(myNumbers[3]); // This will throw an exception

} catch (Exception e) {

System.*out*.println("Something went wrong.");

} finally {

System.*out*.println("I'm from finally block.");

}

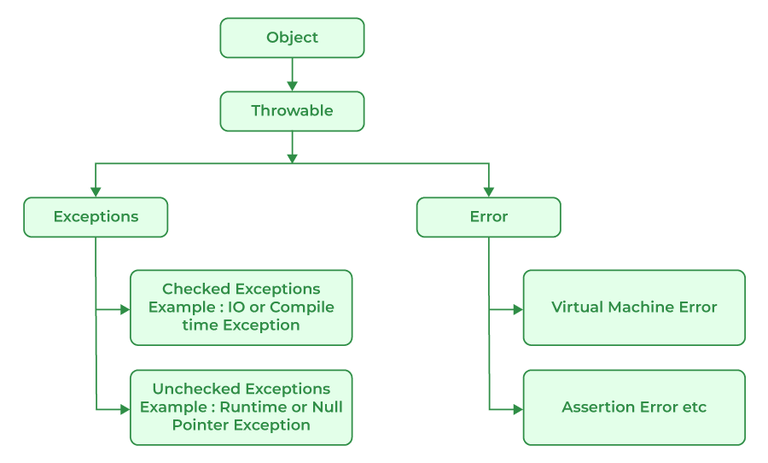
}

}

Output:

Accessing index 3:

Something went wrong.

I'm from finally block.

Task 5

What is the output of the below code… give your reason for the output

public class ExcepTest {

public static void main(String args[]) {

try {

int a[] = new int[2];

int b = 0;

int c = 1/b;

System.out.println("Access element three :" + a[3]);

}

catch (ArithmeticException e) {

System.out.println("ArithmeticException thrown :" + e);

}

catch (ArrayIndexOutOfBoundsException e) {

System.out.println("ArrayIndexOutOfBoundsException thrown :" + e);

}catch (Exception e) {

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

}

System.out.println("Out of the block");

}

}

public class ExcepTest {

public static void main(String args[]) {

try {

int a[] = new int[2]; // Creates array of size 2 (indexes 0 and 1)

int b = 0; // b is initialized to 0

int c = 1/b; // Division by zero will occur here!

System.*out*.println("Access element three :" + a[3]); // This line won't execute

}

catch (ArithmeticException e) {

System.*out*.println("ArithmeticException thrown :" + e);

}

catch (ArrayIndexOutOfBoundsException e) {

System.*out*.println("ArrayIndexOutOfBoundsException thrown :" + e);

}

catch (Exception e) {

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

}

System.*out*.println("Out of the block");

}

}

Output:

ArithmeticException thrown :java.lang.ArithmeticException: / by zero

Out of the block

Reason for this output:

Program Flow:

* Array 'a' is created with size 2
* Variable 'b' is set to 0
* Program attempts to divide 1 by 0 (1/b)
* ArithmeticException is thrown immediately
* The array access statement is never reached

Exception Handling:

The ArithmeticException is caught by the first catch block

Other catch blocks are skipped

Task 5:

Try with Multiple catch blocks …. Execute the below code snippet n display the out .. along with reason..

public class ExcepTest {

public static void main(String args[]) {

try {

int a[] = new int[2];

int b = 0; ⇒ change 0 to 10 and check..

int c = 1/b;

System.out.println("Access element three :" + a[3]);

}

catch (ArrayIndexOutOfBoundsException e) {

System.out.println("ArrayIndexOutOfBoundsException thrown :" + e);

}catch (Exception e) {

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

}

System.out.println("Out of the block");

}

}

public class ExcepTest {

public static void main(String args[]) {

// Scenario 1: with b = 0

try {

int a[] = new int[2];

int b = 0;

int c = 1/b; // ArithmeticException occurs here

System.*out*.println("Access element three :" + a[3]); // Never reached

} catch (ArrayIndexOutOfBoundsException e) {

System.*out*.println("ArrayIndexOutOfBoundsException thrown :" + e);

} catch (Exception e) {

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

}

System.*out*.println("Out of the block");

}

}

Output with b = 0:

Exception thrown :/ by zero

Out of the block

public class ExcepTest {

public static void main(String args[]) {

// Scenario 1: with b = 0

try {

int a[] = new int[2];

int b = 10;

int c = 1/b; // ArithmeticException occurs here

System.*out*.println("Access element three :" + a[3]); // Never reached

} catch (ArrayIndexOutOfBoundsException e) {

System.*out*.println("ArrayIndexOutOfBoundsException thrown :" + e);

} catch (Exception e) {

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

}

System.*out*.println("Out of the block");

}

}

Output with b = 10:

ArrayIndexOutOfBoundsException thrown :java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2

Out of the block

Task 6:

Nested try blocks

public class ExcepTest {

public static void main(String args[]) {

try {

int a[] = new int[2];

System.out.println("Access element three :" + a[2]); // try with a[0] or a[1] ===> and check if control goes to inner try block..

try {

int b = 0;

int c = 1/b;

}catch(Exception e) {

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

}

System.out.println("Access element three :" + a[3]);

}

catch (ArrayIndexOutOfBoundsException e) {

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

}

System.out.println("Out of the block");

}

}

public class ExcepTest {

public static void main(String args[]) {

try {

int a[] = new int[2];

System.*out*.println("Access element three :" + a[2]); // ArrayIndexOutOfBoundsException

try {

int b = 0;

int c = 1/b; // ArithmeticException (never reached in original code)

} catch(Exception e) {

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

}

System.*out*.println("Access element three :" + a[3]);

} catch (ArrayIndexOutOfBoundsException e) {

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

}

System.*out*.println("Out of the block");

}

}

With a[2]:

Exception thrown: java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2

Out of the block

5 min

11.05 to 11.10

—-----------------------------------------------------------------------------------------

=============

Throw and Throws:

Void divide() throws Excepion{

throw …….

}

Void arr2() throws ArrayIndexout….. {

}

Void arr3() throws ArrayIndexout….. {

}

Void method3() throws Exception{

}

psvmain() {

Try{

divide();

arr2();

}catch(ArrayInde…. eobj1){

}catch(Exception eobj2) {

}Finally{

}

}

Task 7:

// Demonstrating how to throw an exception

class MyClass {

static void fun() throws IllegalAccessException

{

System.out.println("Inside fun(). ");

throw new IllegalAccessException("demo exception by fun method");

}

public static void main(String args[])

{

try {

fun();

//method2(); → arrayindex…

//Method3() —> file not found….

}

catch (IllegalAccessException e) {

System.out.println("Caught in main." + e.getMessage());

}

}

}

public class ExceptionDemo {

// Method that throws Exception

static void divide() throws Exception {

int result = 5/0;

throw new Exception("Division by zero error");

}

// Method that throws ArrayIndexOutOfBoundsException

static void arr2() throws ArrayIndexOutOfBoundsException {

int[] arr = new int[2];

System.*out*.println(arr[3]); // This will throw the exception

}

// Another array method

static void arr3() throws ArrayIndexOutOfBoundsException {

int[] arr = new int[3];

System.*out*.println(arr[4]); // This will throw the exception

}

// Method that throws general Exception

static void method3() throws Exception {

throw new Exception("Generic exception from method3");

}

public static void main(String[] args) {

try {

*divide*(); // This will throw Exception

*arr2*(); // This won't be executed due to previous exception

} catch(ArrayIndexOutOfBoundsException eobj1) {

System.*out*.println("Array index error: " + eobj1.getMessage());

} catch(Exception eobj2) {

System.*out*.println("General error: " + eobj2.getMessage());

} finally {

System.*out*.println("Finally block executed");

}

}

}

General error: / by zero

Finally block executed

Task 8:

Custom exceptions: // user defined exception:

// A Class that represents user-defined exception

class MyException extends Exception {

public MyException(String m) {

super(m);

}

}

// A Class that uses the above MyException

public class setText {

public static void main(String args[]) {

try {

// Throw an object of user-defined exception

throw new MyException("This is a custom exception");

}

catch (MyException ex) {

System.out.println("Caught");

System.out.println(ex.getMessage());

}

}

}

============================================

—----------------------------------------------------------------------------------------

==========================================

Task 9:

import java.util.ArrayList;

class Main {

public static void main (String[] args) {

// Creating an ArrayList

ArrayList<Integer> a = new ArrayList<Integer>();

// Adding Element in ArrayList

a.add(1);

a.add(2);

a.add(3);

// Printing ArrayList

System.out.println(a);

}

}

import java.util.ArrayList;

public class ArrayListDemo {

public static void main(String[] args) {

// Creating ArrayList

ArrayList<String> fruits = new ArrayList<>();

// Adding elements

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Orange");

// Printing ArrayList

System.*out*.println("Initial List: " + fruits);

// Adding element at specific index

fruits.add(1, "Mango");

System.*out*.println("After adding Mango at index 1: " + fruits);

// Removing element

fruits.remove("Banana");

System.*out*.println("After removing Banana: " + fruits);

// Checking if element exists

System.*out*.println("Contains Apple? " + fruits.contains("Apple"));

// Getting element at index

System.*out*.println("Element at index 1: " + fruits.get(1));

// Size of ArrayList

System.*out*.println("Size of ArrayList: " + fruits.size());

}

}

Output:

Initial List: [Apple, Banana, Orange]

After adding Mango at index 1: [Apple, Mango, Banana, Orange]

After removing Banana: [Apple, Mango, Orange]

Contains Apple? true

Element at index 1: Mango

Size of ArrayList: 3

Note: You can also create a generic ArrayList

Important Features of ArrayList in Java

ArrayList inherits AbstractList class and implements the List interface.

ArrayList is initialized by size. However, the size is increased automatically if the collection grows or shrinks if the objects are removed from the collection.

Java ArrayList allows us to randomly access the list.

ArrayList can not be used for primitive types, like int, char, etc. We need a wrapper class for such cases.

ArrayList in Java can be seen as a vector in C++.

ArrayList is not Synchronized. Its equivalent synchronized class in Java is Vector.

Task 10

Find the output of the be code snippet..

// Addition, Deletion and Updation of Element

import java.util.\*;

class Main {

public static void main(String args[]){

ArrayList<String> al = new ArrayList<>();

al.add("Prasunamba");

al.add("Meher");

System.out.println("Orignal List : "+al);

al.add(1, "Hello");

System.out.println("After Adding element at index 1 : "+ al);

al.remove(0);

System.out.println("Element removed from index 0 : "+ al);

al.remove("Prasunamba");

System.out.println("Element Prasunamba removed : "+ al);

al.set(0, "K");

System.out.println("List after updation of value : "+al);

}

}

Output:

Original List : [Prasunamba, Meher]

After Adding element at index 1 : [Prasunamba, Hello, Meher]

Element removed from index 0 : [Hello, Meher]

Element Prasunamba removed : [Hello, Meher]

List after updation of value : [K, Meher]