**Name-Soumya Vinod Mudalgi**

**Roll no-24**

**div-B**

**Experiment no-11**

**Experiment name-Implements programs based on try catch and finally block.**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Try** | We specify the block of code that might give rise to the exception in a special block with a “Try” keyword. |
| **Catch** | When the exception is raised it needs to be caught by the program. This is done using a “catch” keyword. So a catch block follows the try block that raises an exception. The keyword catch should always be used with a try. |
| **Finally** | Sometimes we have an important code in our program that needs to be executed irrespective of whether or not the exception is thrown. This code is placed in a special block starting with the “Finally” keyword. The Finally block follows the Try-catch block. |

1.Program using try and catch block.

Input-

class EXCEPTION

{

public static void main(String args[])

{

System.out.println("main method called");

int a=10,b=0,c;

try

{

c=a/b;

System.out.println(+c);

}

catch(Exception e)

{

System.out.println(e);

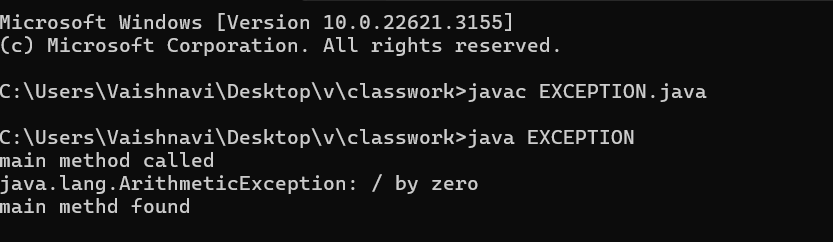
}

System.out.println("main methd found");

}

}

Output-



2.java program using finally keyword.

class EXCEPTION

{

public static void main(String args[])

{

try

{

String s=null;

System.out.println(s.toUpperCase());

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

try

{

System.out.println(10/0);

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

System.out.println("hey");

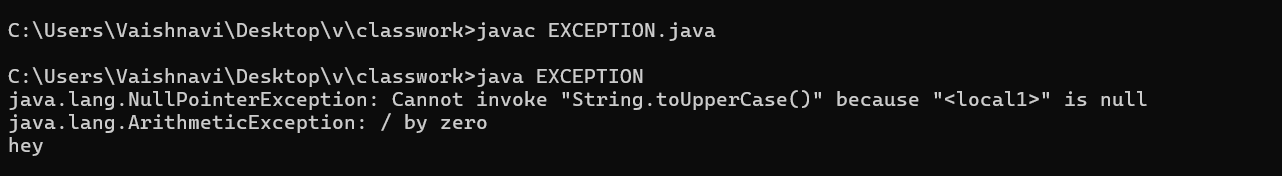
}

}

}

}

Output-



3.java program using nested try.

Input-

class EXCEPTION

{

public static void main(String args[])

{

try

{

try

{

int arr[]={2,4,67};

System.out.println(+arr[5]);

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println(e);

}

System.out.println(10/0);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

Output-

