

NAME : MINAL CHHATRE
ENROLL : 1906016
G3 A

Program 11 Develop programs for implementation of
a)Exception handling
b)User defined exception handling.

CODE: BUILD-IN

```
package Exception_Handling;

public class Sample1 {

    public static void main(String[] args){

        try{

            int a=30,b=0,c;
            c=a/b;
            System.out.println("result = "+c);
        }
        catch(ArithmeticException e){
            System.out.println("Can't divide a number by zero");
        }

        try{
            int num = Integer.parseInt("Minal");
            System.out.println(num);
        }
        catch(NumberFormatException e){
            System.out.println("Number Format Exception");
        }

        try{
            int a[] =new int[5];
            a[7]=5;
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Array index out of bound");
        }
    }
}
```

NAME : MINAL CHHATRE
ENROLL : 1906016
G3 A

OUTPUT:

D:\java>javac -d . Sample1.java

D:\java>java Exception_Handling.Sample1
Can't divide a number by zero
Number Format Exception
Array index out of bound

D:\java>

CODE: USER-DEFINED

```
package Exception_Handling;

class user_defined{

    public static void main(String arg[]){
        try{
            throw new MyException(5);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

class MyException extends Exception{

    int a;
    MyException(int b){
        a=b;
    }
    public String toString(){
        return("Exception number: " +a);
    }
}
```

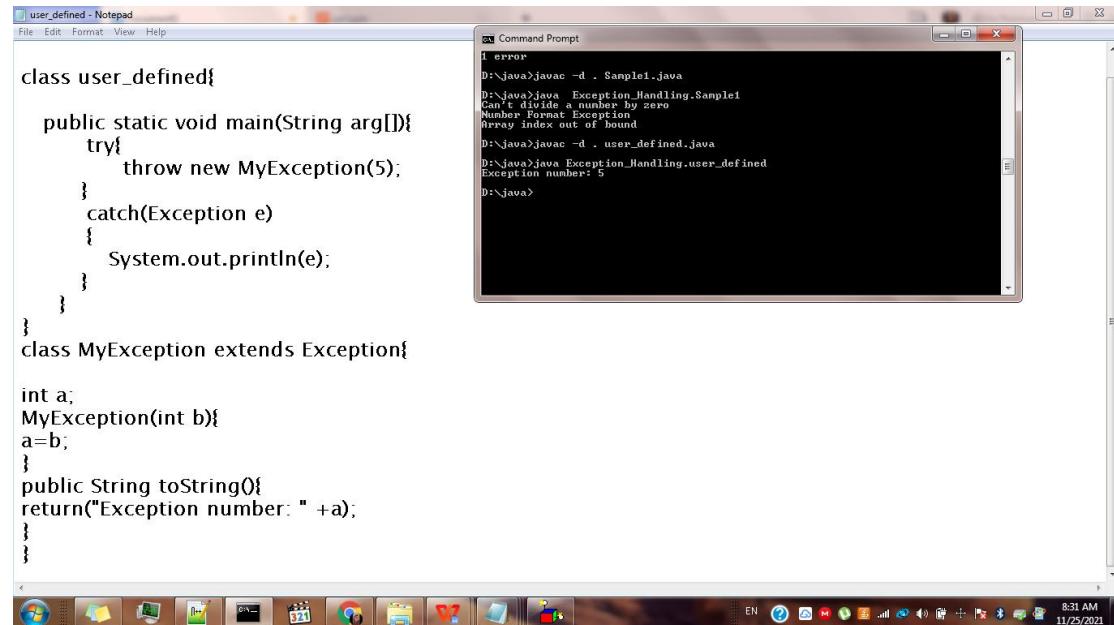
NAME : MINAL CHHATRE
ENROLL : 1906016
G3 A

OUTPUT:

D:\java>javac -d . user_defined.java

D:\java>java Exception_Handling.user_defined
Exception number: 5

D:\java>



The screenshot displays a Windows desktop environment. On the left, a Notepad window titled 'user_defined - Notepad' contains the following Java code:

```
class user_defined{  
    public static void main(String arg[]){  
        try{  
            throw new MyException(5);  
        }  
        catch(Exception e)  
        {  
            System.out.println(e);  
        }  
    }  
}  
class MyException extends Exception{  
    int a;  
    MyException(int b){  
        a=b;  
    }  
    public String toString(){  
        return("Exception number: " +a);  
    }  
}
```

On the right, a Command Prompt window shows the execution of the code. It displays several error messages from previous attempts (e.g., 'Can't divide a number by zero', 'Number Format: Exception', 'Array index out of bound') and the successful output of the current command: 'Exception number: 5'.