

LAB PROGRAM 5

QUESTION: Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

CODE:

```
import java.util.*;
class FatherAgeException extends Exception
{
    public String toString()
    {
        return(" Father's Age cannot be negative");
    }
}
class SonnAgeException extends Exception
{
    public String toString()
    {
        return(" Son's Age cannot be negative");
    }
}
class SonAgeException extends Exception
{
    public String toString()
    {
        return(" Son's age is more than father's age");
    }
}
```

```
class Father
{
    int age;
    Father(int age)
    {
        this.age = age;
    }
    void ThrowingFather()
    {
        try
        {
            if(age < 0)
            {
                throw new FatherAgeException();
            }
        }
        catch (FatherAgeException e)
        {
            System.out.println(e);
        }
    }
}
```

```
class Son extends Father
{
    int sage;
    Son(int fage,int sage)
    {
        super(fage);
        this.sage = sage;
    }
}
```

```

    }
    void ThrowSon()
    {
        try
        {
            if(sage < 0)
            {
                throw new SonnAgeException();
            }
        }
        catch (SonnAgeException e)
        {
            System.out.println(e);
        }
    }
    void ThrowingSon()
    {
        try
        {
            if(sage > age)
            {
                throw new SonAgeException();
            }
        }
        catch (SonAgeException e)
        {
            System.out.println(e);
        }
    }
}

```

```
class Main
{
    public static void main(String args[])
    {
        int fatage,Sonage;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Fathers age");
        fatage = sc.nextInt();
        System.out.println("Enter Son's age");
        Sonage = sc.nextInt();
        Son s = new Son(fatage,Sonage);
        s.ThrowingFather();
        s.ThrowSon();
        s.ThrowingSon();
    }
}
```

WRITTEN CODE

13/01/23

User Defined Exception

```
import java.util.*  
class FatherAgeException extends  
{  
    public String toString()  
    {  
        return ("Father Age cannot  
    }  
}
```

```
class SonAgeException extends  
{  
    public String toString()  
    {  
        return ("son's Age cannot  
    }  
}
```

```
class SonAgeException extends  
{  
    public String toString()  
    {
```

```
void throwingFather ( )
```

```
{
```

```
    try
```

```
    {
```

```
        if (age < 0)
```

```
        {
```

```
            throw new Fa
```

```
        }
```

```
    }
```

```
catch (FatherAgeException
```

```
{
```

```
    System.out.println
```

```
}
```

```
}
```

```
}
```

```
class Son Extends Father
```

```
{
```

```
    int age;
```

```
    son (int fage, int sage,
```

```
    {
```

```
        super(fage);
```

```
        this.age = sage;
```

```
    }
```

```
(void ThrowingFather ( )
```

```

        catch (SonAgeException e)
        {
            System.out.println(e);
        }
    }

    void ThrowingSon()
    {
        long
        &
        if (sage > age)
        {
            throw new SonAge
        }
    }

    }
    catch (SonAgeException e)
    {
        System.out.println(e);
    }
}
}

```

```

class Except
{
    public static void main(
    {
        int fastage, sonage;
    }
}

```



```

s. Throwing Father();
s. Throwing
s. Throwing Son();
s. Throwing Son();

```

```

}
}

```

Output

java Kshiti's java
 java Except

Enter Father's age

-3

Enter Son's age

-1

Father's Age cannot be negative

Son's Age cannot be negative

Son's ~~age~~ is more than father's age

[Signature]

OUTPUT

```
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>javac Java6.java
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
-1
Enter Son's age
23
Father's Age cannot be negative
Son's age is more than father's age
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
23
Enter Son's age
-3
Son's Age cannot be negative
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
12
Enter Son's age
34
Son's age is more than father's age
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
56
Enter Son's age
34
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>
```

