Date :

Date : Dakshith B

Exception Handling

```
import java.util.Scanner;

class WrongAgeException extends Exception{
    public WrongAgeException(String message){
        super(message);
    }
}

class Father{
    private int age;
    public Father(int age) throws WrongAgeException{
        if(age<0){
            throw new WrongAgeException("Age cannot
                be negative");
        }
        this.age = age
    }
    public int sonAge;
    public return age;
}
}
```

Date :

```
class Son extends Father{
    private int sonAge;

    public Son(int FatherAge, int sonAge)
    throws WrongAgeException {
        super(fatherAge);

        if (sonAge >= fatherAge) {
            throw new WrongException("Son's age
            should be less than Father's age);
        }
        this.sonAge = sonAge;
    }
    public int getSonAge(){
        return sonAge;
    }
}

class ExceptionHanding Demo{
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in

        try {
            // System.out.print("Enter Father's Age
            int fatherAge = scanner.nextInt();
```

```
Father fa
System.
+ fath

System.o
int So
Son
System
}

catch (
System

}
finally
{
}
}
}
```

Out

Enter F
50
Enter
25
Fath
Son

```
Father father = new Father ( fatherAge ),
System.out.println (" Father's Age "
  + father.getAge()),

System.out.println(" Enter Son's Age ")
int sonAge = scanner.nextInt ()
Son son = new Son( father.getAge(), sonAge )
System.out.println (" Son's Age " + son.getSonAge(),
}

catch ( WrongAgeException e ) {
  System.out.println (" Exception caught " + e.
      getMessage() );
}
finally {
    scanner.close(),
  }
}
}
```

Output

- Enter Father's age:
  50
  Enter Son's age,
  25
  Father's age: 40
  Son's age: 25

Deekshith B
1BM22CS082

```
import java.util.Scanner;


class WrongAge extends Exception
{
public WrongAge(String str)
{
super(str);
}
}



class Father
{
int fatherAge;
Father(int fatherAge) throws WrongAge
{
this.fatherAge = fatherAge;

if(fatherAge<0)
throw new WrongAge("Father's Age cannot be less than zero");


}
}

class Son extends Father
{
int sonAge;
Son(int fatherAge,int sonAge) throws WrongAge
{
super(fatherAge);
this.sonAge = sonAge;
```

```
if(sonAge<0)
throw new WrongAge("Son's Age cannot be less than zero");

if(sonAge>=fatherAge)
throw new WrongAge("Son's age cannot be greater than or equal to Father's age");

}
}

class AgeExceptionMain
{
public static void main(String[] args)
{
Scanner input = new Scanner(System.in);

System.out.print("Enter father's age: ");
int fatherAge = input.nextInt();

System.out.print("Enter son's age: ");
int sonAge = input.nextInt();

try{
Son son = new Son(fatherAge,sonAge);
System.out.println("Father's and Son's age are valid");

}catch(WrongAge e){
System.out.println("Exception: " + e);
}
}
}
```

Output:

Enter father's age: 50

Enter son's age: 20

Father's and Son's age are valid

Enter father's age: 50

Enter son's age: 60

Exception: WrongAge: Son's age cannot be greater than or equal to Father's age

Enter father's age: -1

Enter son's age: 20

Exception: WrongAge: Father's Age cannot be less than zero

Enter father's age: 0

Enter son's age: 20

Exception: WrongAge: Son's age cannot be greater than or equal to Father's age

Enter father's age: 0

Enter son's age: -1

Exception: WrongAge: Son's Age cannot be less than zero