

NAP that demonstrates handling of exception in inheritance

Base class - father

derived class - son

Father → constructor → takes age

throw exception wrong age if i/p age < 0

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

```
class Father
{
    protected int fatherAge;
```

```
    public Father (int age) throws WrongAge
    {
        if (age < 0)
        {
            throw new WrongAge
                ("Age cant be negative");
        }
        this.fatherAge = age;
        SOP ("father age : " + this.fatherAge);
    }
}
```

~~STREET~~

```
class son extends Father
```

```
{
    private int sonage;
```

```
    public son(int fatherAge, int sonage)
        throws wrongage, exception
```

```
{
    super (fatherage);
```

```
    if (sonage < 0)
```

```
{
        throw new exception ("Not Negative");
    }
```

```
    if (sonage >= fatherage)
```

```
{
        throw new exception ("Not possible");
    }
```

```
    this.sonage = sonage;
```

```
    sop("son's age is set to " + this.sonage);
```

```
    }
}
```

```
public class main
```

```
{
```

```
    psvm
```

```
{
    try
```

```
{ Father f1 = new Father (40);
```

```
    son s1 = new son (f1.fatherage, 25)
```

```
    son s2 = new son (f1.fatherage, 40);
```

```
    son s3 = new son (f1.fatherage, -5);
```

```
    }
```

```
}
```



```

catch (wrongage e)
{
    sop ("Exception occurred");
}
catch (Exception e)
{
    sop ("Exception occurred");
}
}
}

```

O/P:

Father's age : 40

Father's age : 40

Son's age : 25

father's age : 40

Exception occurred : Not possible.