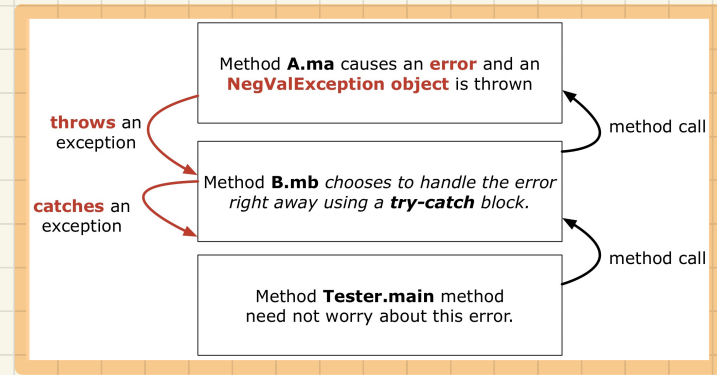


Version 1: B.mb Catches

```
class A {  
    A() {}  
  
    void ma(int i) throws NegValException {  
        if(i < 0) {  
            println("abnormal exec of A.ma");  
            throw new NegValException("Neg Val: " + i);  
        }  
        else {  
            println("normal exec of A.ma");  
        }  
    }  
}
```

```
class B {  
    B() {}  
  
    void mb(int i) {  
        A oa = new A();  
        try {  
            oa.ma(i);  
            println("From B.mb: Calling A.ma did not cause NVE.");  
        }  
        catch(NegValException nve) {  
            println("From B.mb: Calling A.ma caused NVE.");  
        }  
    }  
}
```

```
class Tester {  
    main(...) {  
        int i;  
        B ob = new B();  
        ob.mb(i);  
        println("From Tester.main: After calling B.mb.");  
    }  
}
```



Q1. In B.mb, is calling oa.ma subject to **catch-or-specify** req?

Q2. In Tester.main, is calling B.mb subject to **catch-or-specify** req?

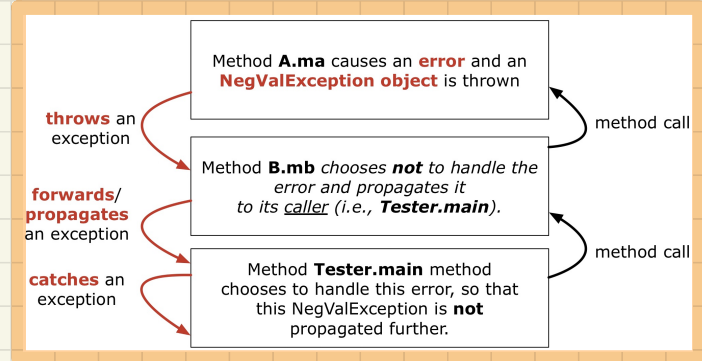
Version 2: B.mb Specifies,

```
class A {  
    A() {}  
  
    void ma(int i) throws NegValException {  
        if(i < 0) {  
            println("abnormal exec of A.ma");  
            throw new NegValException("Neg Val: " + i);  
        }  
        else {  
            println("normal exec of A.ma");  
        }  
    }  
}
```

Tester.main Catches

```
class B {  
    B() {}  
  
    void mb(int i) throws NegValException {  
        A oa = new A();  
        oa.ma(i);  
        println("From B.mb: Calling A.ma did not cause NVE.");  
    }  
}
```

```
class Tester {  
    main(...) {  
        int i;  
        B ob = new b();  
        try {  
            ob.mb(i);  
            println("Tester.main: Calling B.mb did not cause NVE.");  
        }  
        catch(NegValException nve) {  
            println("Tester.main: Calling B.mb caused NVE.");  
        }  
    }  
}
```



Q1. In B.mb, is calling oa.ma subject to **catch-or-specify** req?

Q2. In Tester.main, is calling B.mb subject to **catch-or-specify** req?

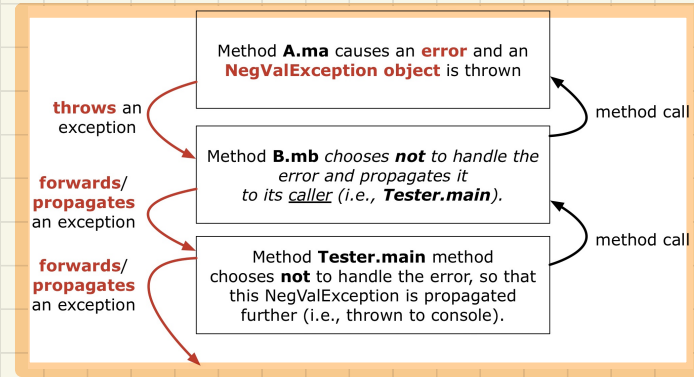
Version 3: B.mb Specifies,

```
class A {  
    A() {}  
  
    void ma(int i) throws NegValException {  
        if(i < 0) {  
            println("abnormal exec of A.ma");  
            throw new NegValException("Neg Val: " + i);  
        }  
        else {  
            println("normal exec of A.ma");  
        }  
    }  
}
```

Tester.main Specifies

```
class B {  
    B() {}  
  
    void mb(int i) throws NegValException {  
        A oa = new A();  
        oa.ma(i);  
        println("From B.mb: Calling A.ma did not cause NVE.");  
    }  
}
```

```
class Tester {  
    main(...) throws NegValException {  
        int i;  
        B ob = new b();  
        ob.mb(i);  
        println("Tester.main: Calling B.mb did not cause NVE.");  
    }  
}
```



Q1. In B.mb, is calling oa.ma subject to **catch-or-specify** req?

Q2. In Tester.main, is calling B.mb subject to **catch-or-specify** req?

Catch-or-Specify Requirement: Call Stack

Q1. Origin of Exception:

Q2. Methods subject to CoS Req:

Q3. Methods free from CoS Req:

Q4. Extreme Case 1 (exception handled earliest):

Q5. Extreme Case 2 (exception never handled):

$C1.m1$

$C2.m2$

$C3.m3$

$\dots$

$C_i.m_i$

$C_{i+1}.m_{i+1}$

$\dots$

$C_{n-2}.m_{n-2}$

$C_{n-1}.m_{n-1}$

$C_n.m_n$