

.....  
Vorname, Name, Matrikelnummer

1. What is wrong with this code?

```
public void Foo () {  
    try {  
        lock() // lock some resource  
        // open some resource  
        // try to change some things  
        // fool around a bit  
  
    }  
    catch (Exception e) {  
    }  
    catch (ConcurrentModificationException e) {  
  
    System.out.println(„ bad stuff going on today!“);  
    }  
    finally {  
        unlock();  
        return; // kein return  
                // bei finally  
    }  
}
```

*// Kein Inhalt, fängt schon alle Fehler ab*

*// wird nicht aufgerufen durch  
// over catch  
// sollte log. error sein  
// nicht aussagekräftig  
keine Meldung von Fehler*

*↳ throw Exception  
↳ mitteilung, unterdrückung*

2. What will be the output of the program?

```
public class TestException  
{  
    public static void badCall()  
    {  
        System.out.print("throwing it ");  
        throw new RuntimeException();  
    }  
    public static void main(String [] args)  
    {  
        try  
        {  
            System.out.print("hello ");  
            badCall();  
        }  
        catch (Exception re )  
        {  
            System.out.print("caught ");  
        }  
        finally  
        {  
            System.out.print("finally ");  
        }  
        System.out.println("after ");  
    }  
}
```

*// Fehlerzeugung*

*// laut*

*// throw Exception*

- Ausgabe: hello | throwing it | caught | finally | after

### 3. Output?

```
public class TestException1
{
    public static void main(String [] args)
    {
        try
        {
            badMethod();
            System.out.print("A");
        }
        catch (Exception ex)
        {
            System.out.print("B");
        }
        finally
        {
            System.out.print("C");
        }
        System.out.print("D");
    }
    public static void badMethod()
    {
        throw new Error();
    }
}
```

(Programm stürzt ab)  
↳ Error wird geworfen  
Thread wird abgebrochen  
→ Error wird nicht gefangen  
Ausgabe: C

### 4. Make it compile!

```
public class TestException2
{
    class TestException extends Exception {} // inner class
    public void runTest() throws TestException {}

    public void test()
    { try {} catch {} }
    runTest();
}
}
```

### 5. When should you re-throw a caught exception?

→ Wenn man Fehler nicht "heilen" kann

### 6. A banking software detects, that a certain customer ID is not in the database.

Is this a) a system exception, b) a custom exception, c) no exception.

→ eigene Exception

### 7. Was ist der Vorteil von Exceptions gegenüber dem Auswerten von Fehlerwerten im Return?

– Exceptions sind einfacher zu handeln / verstehen