Session: Exceptions in Java

Toni, Lisa, Lisa, Johannes
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 {
                return;
        }
}
```

→ empty braces, nothing happens

switch order

use log instead of sysout, more informative text

shouldn't be inside finally {}

- second exception is never caught as the first (more general one) catches all exceptions
  → always start with the most specific exception!
  ↳ more helpful when debugging

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 ");
    }
}
```

→ reached end of file while parsing

→ after adding brace: "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();
        }
}
```

CException in thread "main" java.lang.Error
 at aufg.TestException1.badMethod(TestException1.java:24)
 at aufg.TestException1.main(TestException1.java:9)

4. Make it compile!

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

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

↳ (add exception to method signature)

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

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

if the catch-block is unable to handle it

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.

Depends on the implementation but b) would probably make the most sense ~ sth like "customerNotFoundException"

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

Errors are already categorized when they occur and can be handled right away inside the code