Module 1 - Lecture 16

Exception Handling & File I/O Part 1

What's wrong with this?

```
String storeName;
double sum = 0.0;

for(int i = 0; i <= array.length; i++) {
    sum += array[i];
}

System.out.println("You spent " + sum + " at " + storeName);
```

What might be wrong with this?

```
public void printSum(String storeName, double[] array) {
    double sum = 0.0;

    for(int i = 0; i < array.length; i++) {
        sum += array[i];
    }

    sysout(“You spent “ + sum + “ at “ + storeName);
}
```

Compile vs Runtime Errors

- A **compile-time error** occurs when there is a syntactical error or the compiler identifies code that cannot execute (e.g. method does not exist, accessibility not public, redefined constant, type mismatch)

- A **run-time error** occurs while the program is executing. For example, the program tries to access beyond the bounds of a collection, access a restricted file, parse an invalid value, etc.

  Run-time errors can be dealt with using **exception handling**

Exception Handling

Exception Handling

Exception handling is the process of responding to exceptional errors in the programming. This processing often changes the flow of the program so that it can recover.

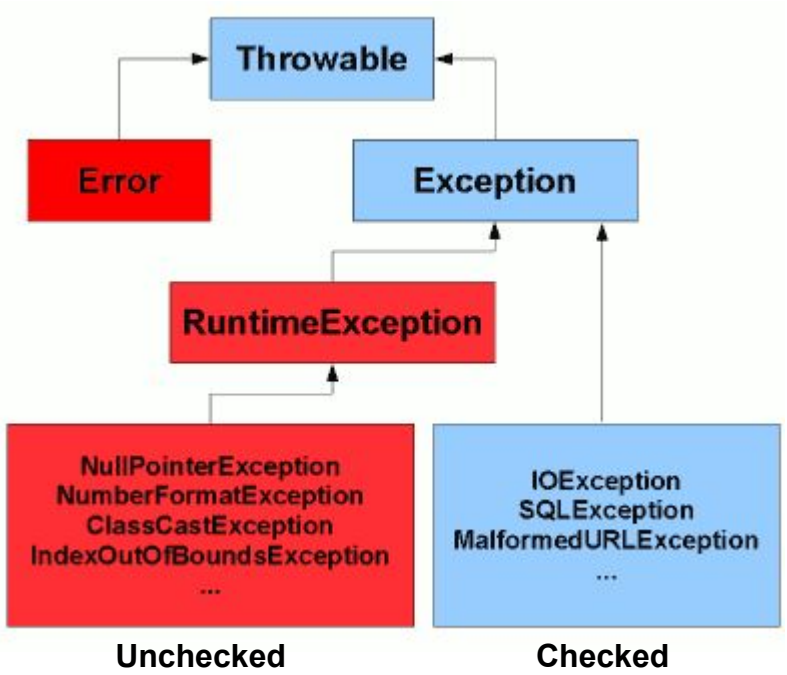
Examples:

- A file is deleted while the program is executing
- The network shuts down while calling an API
- A negative value is passed into a method that can't handle negative values (typically not handled at compile time, but by throwing an InvalidArgumentException)

Checked vs. Unchecked

Checked exceptions are exceptional cases that can occur while your code is running. The compiler will ensure that the developer handles these appropriately. You must either catch or rethrow them.

Unchecked exceptions are exceptions that are preventable. They occur at runtime.



Let's Code!

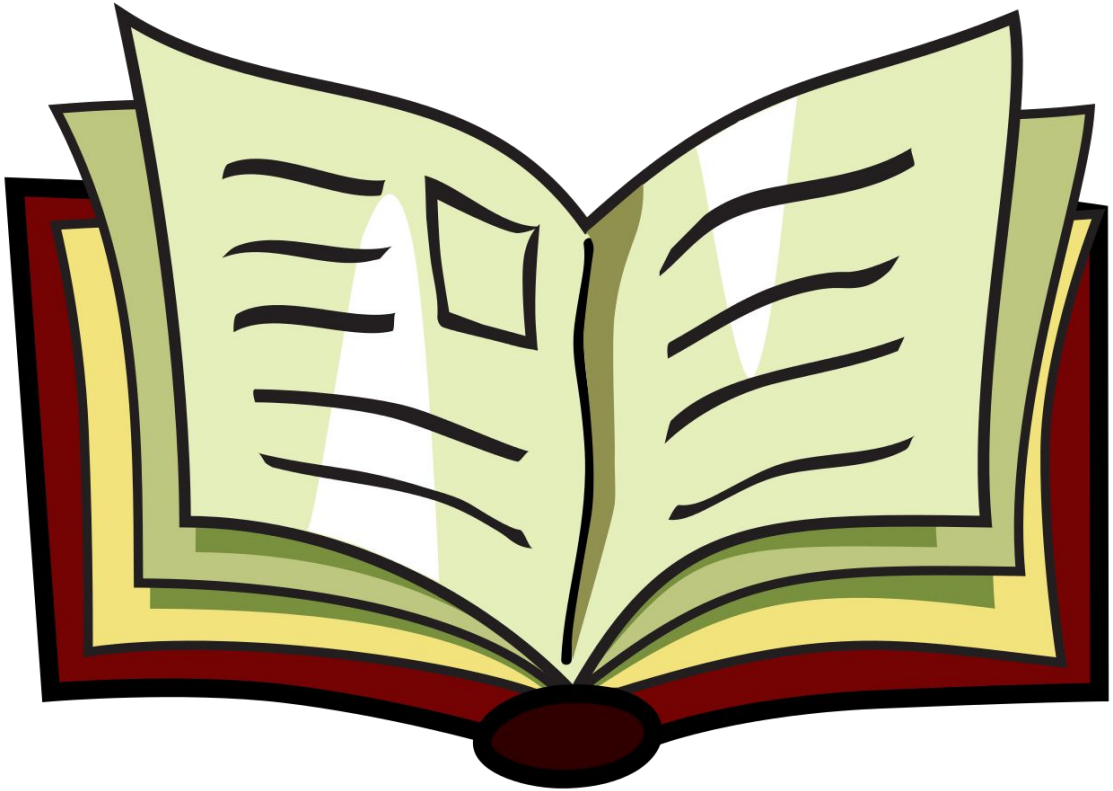
Recap on Exception Handling

Important Points of Exception Handling

- try/catch/finally structure
- Stack traces identify the path leading up to an exception
- Exceptions are classes that inherit from java.lang.Exception
- In Java you must handle checked exceptions by either rethrowing them or catching them

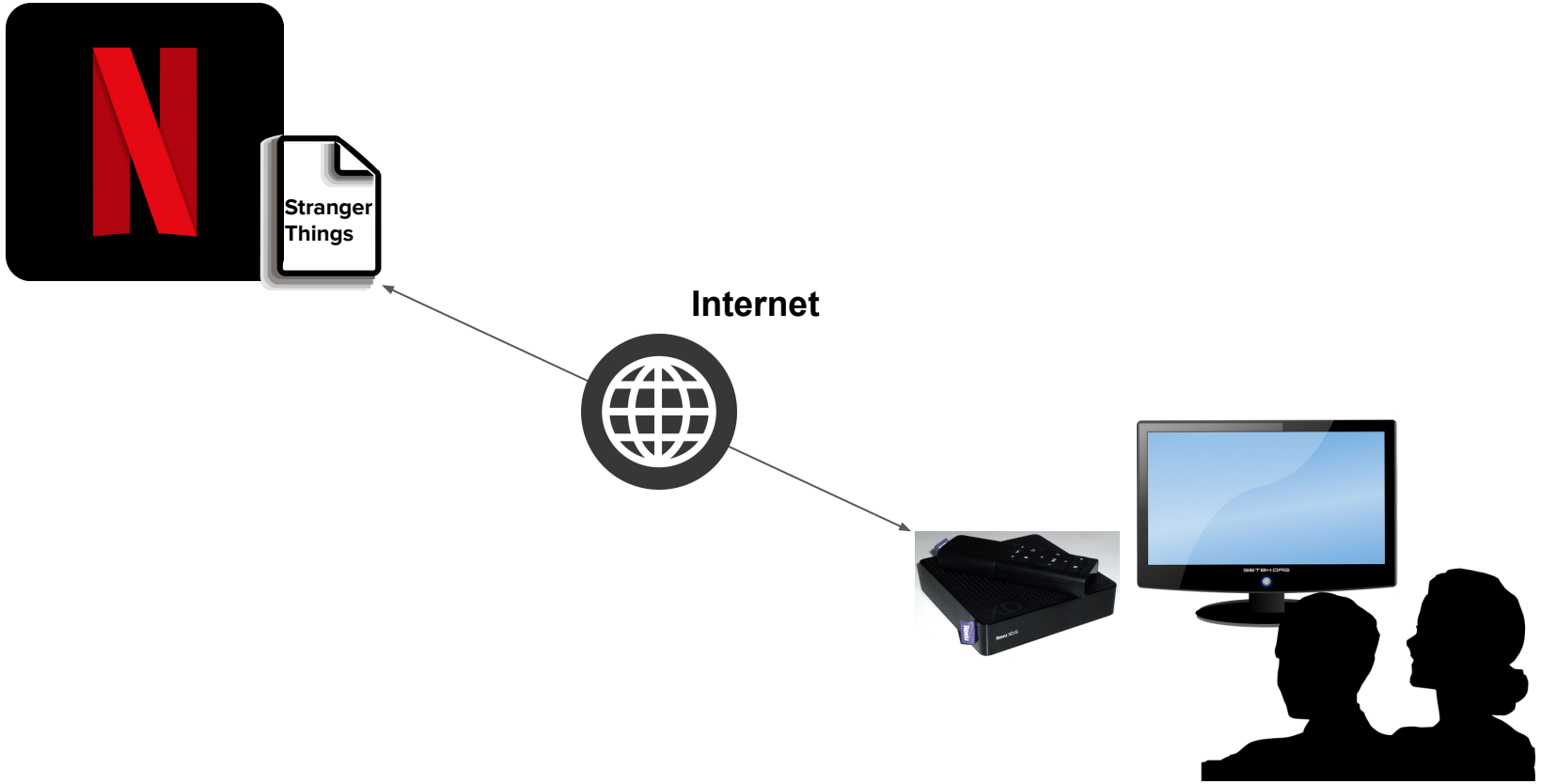
- (Advanced) You can make your own exceptions to handle control flow in the application

File I/O - Reading

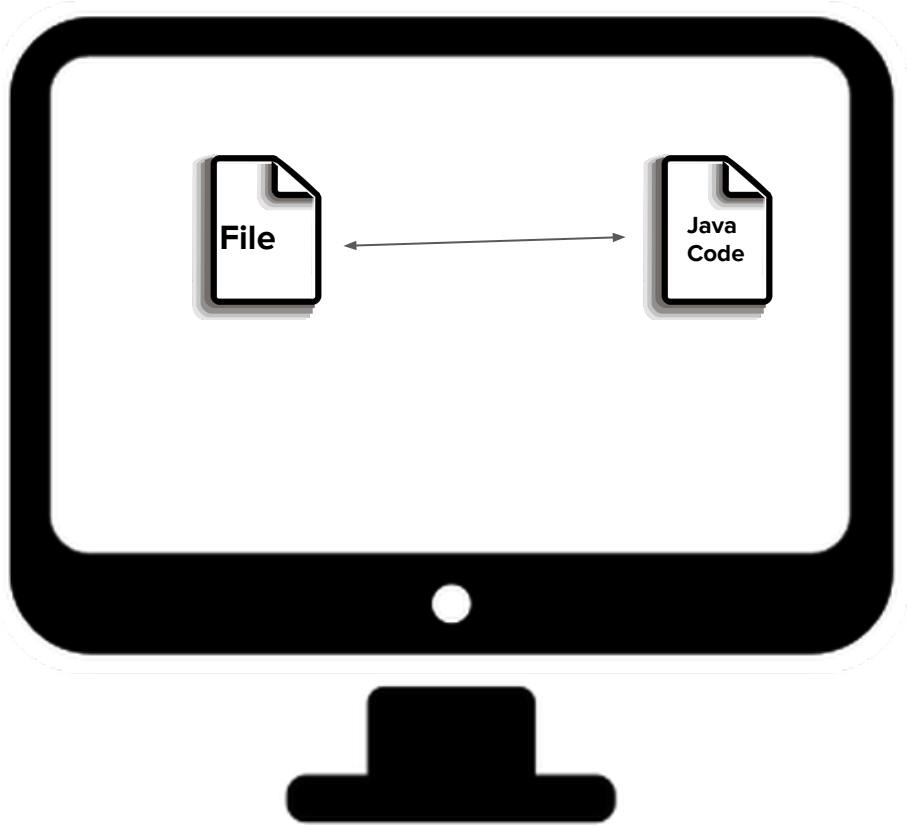



NETFLIX

Streaming



Streaming simplified



What is a stream?

A **stream** refers to a sequence of bytes that can read and write to some sort of backing data store.

Streams have an end-of-file marker or end-of-stream marker to indicate when the program reaches the end of the stream.

A stream is like an assembly line, where you process each thing as it comes through, in order.

In Java, there are many ways to stream a file. We will use the Scanner.

Can I read an entire file?

Methods exist to read all data in a file with one line of code. They pull the entire file into memory though. Sometimes we don't want to do that, especially for large files. This is equivalent of sitting to watch a NetFlix movie and waiting for the entire movie to load before you start watching it.

Let's Code!

QUESTIONS?