

Exceptions

Overview

- “An exception is an object that defines an unusual or erroneous situation.”
- Examples
 - Divide by 0
 - Array index out of bounds
 - File cannot be found
 - Follow a null reference

Uncaught Exceptions

```
int a = 5;  
int b = 0;  
int c = a/b;
```

```
Exception in thread "main" java.lang.ArithmeticException: /  
by zero  
at ExceptionsExamples.main(ExceptionsExamples.java:12)
```

Uncaught Exceptions

- Call stack trace
 - info about where the exception occurred
 - info about all methods that were called to get to the method where the exception occurred

```
Exception in thread "main" java.lang.ArithmeticException: / by zero  
at ExceptionsExamples.caller2(ExceptionsExamples.java:20)  
at ExceptionsExamples.caller1(ExceptionsExamples.java:12)  
at ExceptionsExamples.main(ExceptionsExamples.java:8)
```

try/catch

```
try {  
    //statements that may throw an exception  
} catch(Exception_Type name) {  
    //what to do in case of exception  
    //Exception_Type  
} catch(Another_Type another_name) {  
    //what to do in case of exception Another_Type  
}
```

Propagation

```
void divider() throws ArithmeticException {  
    int a = 5;  
    int b = 0;  
    int c = a/b;  
}
```

Exception Class Hierarchy

- Throwable
 - Error
 - Exception
 - IOException
 - EOFException, FileNotFoundException
 - ClassNotFoundException
 - RuntimeException
 - ArithmeticException
 - ClassCastException
 - NullPointerException

Designing/Throwing Own Exception

- Create a class that extends Exception
 - OutOfRangeException

```
if out of range {  
    throw new OutOfRangeException("message");  
}
```