

Exceptions

Atul Prakash

Exceptions

- Exceptions enable you to write readable code and handle errors in a consistent way
- You separate the code that describes what you want to do from the code that handles problems
- Exception handling is the only official way that Java reports errors

Exceptions: try catch block

- Called a try block because you “try” method calls

```
try {  
    // Application code goes here..  
}  
catch(ExceptionType1 e1) {  
    // Handle exceptions for Type1  
}  
catch(ExceptionType2 e2) {  
    // Handle exceptions for Type2  
}  
catch(ExceptionType3 e3) {  
    // Handle exceptions for Type3  
}
```

Exception Syntax

```
try {  
    // Your code goes here..  
}  
catch (exceptiontype arg) {  
    // Your error handling code goes here..  
}
```

```
try {  
    int result = x/0;  
    System.out.println("result: " + result);  
}  
catch (ArithmeticException e) {  
  
    System.out.println("caught an exception: " + e);  
  
}
```

<http://java.sun.com/j2se/1.4.2/docs/api/java/lang/ArithmeticException.html>

```
// OK
try {
    int result = x/y;
    System.out.println("result: " + result);
}
catch (Exception e) {
    System.out.println("caught an exception: " + e);
}

// Preferred: identifies the cause better.
try {
    int result = x/y;
    System.out.println("result: " + result);
}
catch (IOException e) {
    System.out.println("caught an i/o exception: " + e);
}
catch (ArithmeticException e) {
    System.out.println("caught an math exception: " + e);
}
```

Uncaught Exceptions

- If an exception is not caught in a function:
 - The function terminates and the exception passes to the caller
 - This keeps happening till the exception is caught in a caller.
 - If no one catches, the program terminates.

Exception Handling

```
void f(int x, int y) {  
    try {  
        int g(x, y);  
    }  
    catch (ArithmeticException e) {  
        System.out.println("caught an arithmetic exception: " + e);  
    }  
}  
  
void g(int x, int y) {  
    int result = x/y; // If y is 0, exception thrown.  
    System.out.println("result: " + result);  
}
```

