

“catching” an error

```
try
{
  x = Integer.parseInt( userString );
}
catch (Exception e)
{
  System.out.println("Not an integer!");
}
```

try...catch

- “Tries” the try-bracketed statement.

```
try
{
  // whatever is here
}
```

- Stops at an error. Does not complete.
- Error info is placed into the object: “e”, of the Exception class - can decipher ALL exceptions

```
catch (Exception e)
{
  // do something here
}
```

Exception class

catch (Exception e)

- declares an object e of the Exception class
- there are methods to be used

```
catch (Exception e)
{
  e.printStackTrace(); // where the error was
  System.out.println("input was" + e.getMessage() );
  // what input caused error
}
```

Other Exception methods()

Exception e

e.getMessage() - what the user typed that generated the error

e.toString() - the computer's description of the message

e.printStackTrace() - list all the methods from main, to the one that generated the error

catch different types

```
try {
    myClass myObject = new myClass();
    myObject.myMethod( );
    // method 2( )
    // method 3( )
}
catch (NullPointerException e)
{
}
catch (DivideByZeroException e)
{
    System.out.println("Can't divide by zero");
}
```

different types

- NullPointerException - method called in try, was supposed to return an object (like Color), but didn't (user hit cancel in JColorChooser)
- ArrayIndexOutOfBoundsException - - method called in try, accessed index greater than the array definition.

different types

- NumberFormatException - - method called in try, like parseInt, tried to convert a string to a number, and the string wasn't a digit.
- ArithmeticException - method called in try, did something arithmetically wrong - like divide by zero.

different types

- AWTException - method called in try, attached a handler object that was inappropriate for the component (a MouseListener to a JButton, or an object not of an ActionListener class).
- DivideByZeroException - method called in try, did a divide by zero.

“throwing” an exception

```
public myClass
{
public double myMethod( int q ) throws DivideByZeroException
{
if (q == 0)
throw new DivideByZeroException( );
else
return(10/q);
}
}
```
