Mainstream Java

Handling Exceptions

Handling Exceptions: Try/Catch

```
URL myURL = new URL(someString);
//code that uses myURL
//e.g., connects to it and reads data
```

- Have code which might throw an exception
  - Example: MalformedURLException

Handling Exceptions: Try/Catch

```
try {
    URL myURL = new URL(someString);
    //code that uses myURL
    //e.g., connects to it and reads data
}
```

- Try-block:
  - Code that might throw an exception

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Catch block
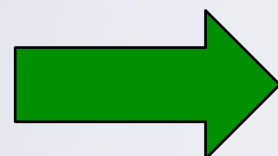 - Specifies what to do if exception is thrown

Handling Exceptions: Try/Catch

```
try {
  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Normal execution (no exception)

  - someString = "http://www.duke.edu"

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```
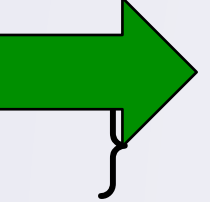
- Normal execution (no exception)
  - someString = "http://www.duke.edu"

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Normal execution (no exception)

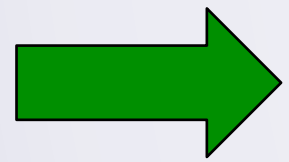 - someString = "http://www.duke.edu"

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}

catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Normal execution (no exception)

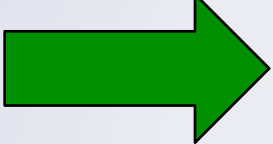 - someString = "http://www.duke.edu"

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

→

- Normal execution (no exception)

  - someString = "http://www.duke.edu"

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Exception is thrown

  - someString = "blahblahblah"
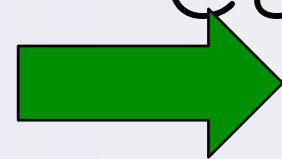
Handling Exceptions: Try/Catch

```
try {

URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Exception is thrown
  - someString = "blahblahblah"
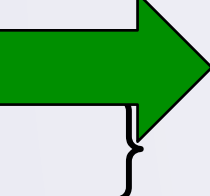
Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data

}
catch (MalformedURLException mfe) {
  //code to handle the error
}
```

- Exception is thrown
    - someString = "blahblahblah"
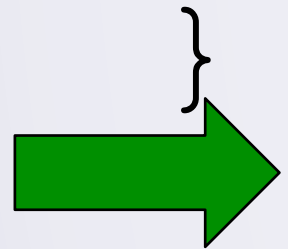
Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data

}
catch (MalformedURLException mfe) {
    //code to handle the error

}
```

- Exception is thrown

  - someString = "blahblahblah"

Handling Exceptions: Try/Catch

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
```

- Exception is thrown

  - someString = "blahblahblah"

Handling Exceptions: Finally

```
try {

  URL myURL = new URL(someString);
  //code that uses myURL
  //e.g., connects to it and reads data
}
catch (MalformedURLException mfe) {
    //code to handle the error
}
 finally {
    //code to do whether exception or not
 }
```

- Finally: **always** executed