Catching Exceptions

Uncaught Exceptions

- If an exception is thrown and not caught the program ends
- A message is printed that contains:
  - The name of the exception
  - The "message" associated with the exception
  - The program stack

```
Exception in thread "main" java.lang.NullPointerException
        at inherShapes.Shape.toString(Shape.java:16)
        at inherShapes.Rectangle.toString(Rectangle.java:20)
        at java.base/java.lang.String.valueOf(String.java:2951)
        at inherShapes.TestShapes.main(TestShapes.java:15).
```

The try/catch statement Syntax

```
try { tryBlock }
catch( exceptionClass name) { catchBlock }
catch( exceptionClass name) { catchBlock }
…
finally { finalBlock }
```

- *tryBlock* any Java instructions which might throw an exception
- *exceptionClass name* declaration of the local name of the caught exception within the catch block
- *catchBlock* Java instructions executed when an exception is thrown (and caught)
- *finalBlock* java instructions executed whether or not an exception is caught

Program Stack Example

- Java invokes TestShapes main method
  - TestShapes.main invokes Rectangle.toString()
    - Rectangle.toString invokes super.toString which is Shape.toString
      - Shape.toString throws an exception

- Program Stack:
  - Shape.java:16 – null pointer exception
  - Rectangle.java:20 – invocation of super.toString()
  - TestShape.java:15 – implicit invocation of Rectangle.toString()