

- Quiz 2 Monday
- Homework 3 due Monday

---

## Flashback

- Exceptions are handled within **try** blocks by handlers defined in **catch** blocks.
  - Catching an exception restores normal program flow, starting inside the **catch** block.
  - Failing to catch an exception exits a program.
  - Checked exceptions (those not a subclass of **RuntimeException**) must be caught or else declared in the **throws** clause of the method.
-

In-class lab exercise #2

# Creating Subclasses

Create a model class using information learned about exceptions, javadoc, abstract classes, inheritance, etc.

## ②

# Creating Subclasses

Extend the base class **Shape** (downloaded from the course website) :

Follow the example of the **Shape** class in including javadoc comments and exception throws for out-of-bounds parameters.

1. Create a subclass of **Shape** called **Ellipse**. It should have two additional fields for the width and height (for the major and minor axes).
2. Create a **Ellipse** constructor which reuses the **Shape** constructor, then initializes the extra fields.
3. Create accessors (“getters”) and mutators (“setters”) for the new fields.
4. Define the abstract method **area()**. (The area of an ellipse is  $\pi hw / 4$ .) A predefined constant **Math.PI** is available for your use.
5. Override the **toString()** method from **Shape**, but reuse its implementation in your output.
6. Create a main method that creates a **Ellipse** and prints it to the console.
7. Compile and run your program.
8. Run the javadoc compiler on the classes (**javadoc \*.java**)
9. Open the file **index.html** in a browser and examine the result.