Topics: Writing subclasses; graphics; `static` components

Reading: (T) Sec 1.4-1.7, Sec 13.0-13.2.1

Writing a subclass

Keyword: `extends`
Example class header: `public class MyFrame extends JFrame`

A subclass builds on an existing class. In the subclass, add the needed functionality that is not in the super class. The subclass has direct access to all the `public` components of the super class—the components are *inherited*.

Creating an object of a subclass

```
new classname()
```

Example: `MyFrame f= new MyFrame();`

Examples

Edit class `MyFrame`:

- Keep track of the length of the short side of the frame before the squaring operation
- Draw a square in the frame
- Write a method to take the absolute value of a passed integer (input argument)
- Keep track of the number of times method `absolute` has been called

OOP ideas

- Aggregate variables/methods into an abstraction (a class) that makes their relationship to one another explicit
- Objects (instances of a class) are self-governing (protect and manage themselves)
- Hide details from client, and restrict client's use of the services
- Allow clients to create/get as many objects as they want

```
import javax.swing.*;


/** Square frames */
public class SquareFrame extends JFrame {




    /** make this window a square */
    public void makeSquare() {


        int dim= Math.max(this.getHeight(), this.getWidth());
        this.setSize(dim,dim);
    }









}
```