Design by Prototype

Programmers often approach a problem by creating a series of prototypes. Each prototype is intended to make progress toward the desired final program.

Example: an email program

- Prototype 1: only allows the user to read email messages.
- Prototype 2: allows the user to read and write text-only messages.
- Prototype 3: adds support for attachments to Prototype 3.
- Prototype 4: adds facilities for an address book and message flagging.
- Prototype 5: has better user interface in response to customer testing.

Program Requirements:
Create the following window.

How would you design this program using a prototyping approach?
Debugging

The process of identifying and correcting program errors is called **debugging**. The compiler captures syntax errors. However, programmers need to capture their own logic errors.

**Two programmer “tricks” for locating logic errors**

- **comment out code**
  
  bracket selected regions of code with /* ... */ and observe the behavior of the resulting program.

- **intersperse System.out.printlns**
  
  include println instructions to display selected parts of the state and analyze the output of the resulting program.

Commenting out code

Consider the following erroneous attempt at a canoe prototype.

**Desired window**

![Desired window](image)

**Actual window**

![Actual window](image)

The faulty code is on the next page...
```java
import java.awt.Color;
import javax.swing.JFrame;

public class Driver {
    private JFrame theWindow;
    private Rectangle hull;
    private Oval bow, stern, cover;

    /** post: a window is constructed and a canoe on a white background displayed. */
    public Driver() {
        theWindow = new JFrame( "Canoe" );
        theWindow.setBounds( 30, 30, 280, 240 );
        theWindow.setLayout(null);
        theWindow.setVisible(true);
        theWindow.getContentPane().setBackground( Color.white );

        hull = new Rectangle( 40, 90, 200, 51 );
        hull.setBackground( Color.lightGray );
        theWindow.add( hull, 0 );

        bow = new Oval( 20, 80, 40, 60 );
        bow.setBackground( Color.lightGray );
        theWindow.add( bow, 0 );

        stern = new Oval( 220, 80, 40, 60 );
        stern.setBackground( Color.lightGray );
        theWindow.add( stern, 0 );

        /*
        cover = new Oval( 20, 6, 240, 100 );
        cover.setBackground( Color.white );
        theWindow.add( cover );
        */
        theWindow.repaint();
    }
}
```

**What if we comment out the code involving cover?**

Commenting out code is good for narrowing down the error’s source. Be careful not to attempt nesting of comments.
System.out.println

Java includes a special method (println) that can be used to output the state of any object.

System.out.println( objectReference );

Desired window

Actual window

The faulty code is on the next page...

```java
import java.awt.Color;
import javax.swing.JFrame;

public class Driver {
    private JFrame theWindow;
    private Rectangle hull;
    private Oval bow, stern, cover;

    /** post: a window is constructed and a canoe on a white background displayed. */
    public Driver() {
        theWindow = new JFrame( "Canoe" );
        theWindow.setBounds( 30, 30, 280, 240 );
        theWindow.setLayout(null);
        theWindow.setVisible(true);
        theWindow.getContentPane().setBackground( Color.white );

        hull = new Rectangle( 40, 90, 200, 51 );
        hull.setBackground( Color.lightGray );
        theWindow.add( hull, 0 );

        bow = new Oval( 20, 80, 40, 60 );
        bow.setBackground( Color.lightGray );
        theWindow.add( bow, 0 );

        stern = new Oval( 20, 80, 40, 60 );
        stern.setBackground( Color.lightGray );
        theWindow.add( stern, 0 );

        cover = new Oval( 20, 6, 240, 100 );
        cover.setBackground( Color.white );
        theWindow.add( cover, 0 );

        theWindow.repaint();
    }
}
```
import java.awt.Color;
import javax.swing.JFrame;

public class Driver {
    private JFrame theWindow;
    private Rectangle hull;
    private Oval bow, stern, cover;

    /** post: a window is constructed and a canoe on a white background displayed. */
    public Driver() {
        theWindow = new JFrame( "Canoe" );
        theWindow.setBounds( 30, 30, 280, 240 );
        theWindow.setLayout(null);
        theWindow.setVisible(true);
        theWindow.getContentPane().setBackground( Color.white );
        hull = new Rectangle( 40, 90, 200, 51 );
        hull.setBackground(Color.lightGray);
        theWindow.add( hull, 0 );
        System.out.println( hull );
        bow = new Oval( 20, 80, 40, 60 );
        bow.setBackground(Color.lightGray);
        theWindow.add( bow, 0 );
        System.out.println( bow );
        stern = new Oval( 20, 80, 40, 60 );
        stern.setBackground(Color.lightGray);
        theWindow.add( stern, 0 );
        System.out.println( stern );
        cover = new Oval( 20, 60, 240, 100 );
        cover.setBackground( Color.white );
        theWindow.add( cover, 0 );
        theWindow.repaint();
    }
}