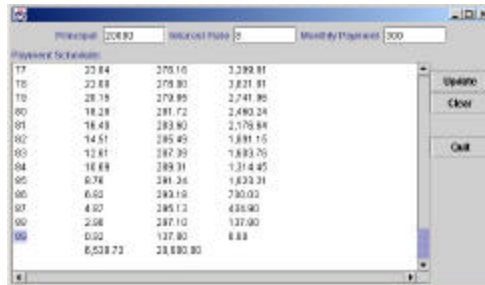


## Example 2: Text components

- **JTextField**
  - single line for input
  - main interesting event is pushing Return
- **JTextArea**
  - multiple lines; can add scrolling
  - edit in place
  - change size and font for whole area but not parts
  - fancier JTextComponents for editing, display of different sizes and fonts, HTML, etc.



The screenshot shows a Java Swing window titled "Payment Schedule". It has three text input fields at the top: "Principal" with the value "20000", "Interest Rate" with the value "5", and "Monthly Payment" with the value "300". Below these fields is a table with the following data:

	Principal	Interest Rate	Monthly Payment
T7	22.84	278.10	3,298.81
T8	22.88	278.80	3,821.81
T9	28.78	278.89	2,741.80
80	18.28	281.72	2,482.24
81	16.48	283.80	2,178.84
82	14.51	285.49	1,881.15
83	12.61	287.39	1,603.76
84	10.88	289.31	1,344.45
85	9.38	291.26	1,103.21
86	8.02	293.18	790.02
87	6.82	295.12	498.80
88	5.78	297.10	227.80
89	4.92	297.80	8.88
90	4.23	298.00	
	6,528.73	28,880.80	

On the right side of the window, there are three buttons: "Update", "Clear", and "Quit".

## Example 2 code excerpts

```
class mtg extends JFrame implements ActionListener {
    JLabel lprin = new JLabel("Principal ");
    JTextField tprin = new JTextField(7);
    JLabel lrate = new JLabel("Interest Rate");
    JTextField trate = new JTextField(7);
    JLabel lmpay = new JLabel("Monthly Payment");
    JTextField tmpay = new JTextField(7);

    JLabel lsched = new JLabel("Payment Schedule:");
    JTextArea tpay = new JTextArea(15, 45);

    JButton update = new JButton("Update");
    JButton clear = new JButton("Clear");
    JButton quit = new JButton("Quit");

    public static void main(String[] args) {
        mtg m = new mtg();
    }
}
```

## Example 2, page 2

```
mtg() {
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
    // top row of entry boxes
    JPanel ptop = new JPanel();
    ptop.add(tprin);
    ptop.add(tpin);
    ptop.add(trate);
    ptop.add(trate);
    ptop.add(tpay);
    ptop.add(tpay);
    tpin.setToolTipText("Enter principal amount");
    trate.setToolTipText("Enter yearly interest rate ...");
    tpay.setToolTipText("Enter monthly payment");

    // text area for payment schedule
    JScrollPane jsp = new JScrollPane(tpay,
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
        JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
    JPanel pctr = new JPanel(new BorderLayout());
    pctr.add(jsched, BorderLayout.NORTH);
    pctr.add(jsp, BorderLayout.CENTER);
}
```

## Example 2, page 3

```
// buttons on right
JPanel pr = new JPanel(new GridLayout(0,1));
pr.add(new JLabel()); // spacer
pr.add(update);
pr.add(clear);
pr.add(new JLabel()); // spacer
pr.add(quit);
JPanel pright = new JPanel(new BorderLayout());
pright.add(pr, BorderLayout.NORTH); // pack at top

update.addActionListener(this);
clear.addActionListener(this);
quit.addActionListener(this);
update.setToolTipText("Update payment schedule");
clear.setToolTipText("Clear payment schedule");

// overall layout
Container cp = getContentPane();
cp.add(ptop, BorderLayout.NORTH);
cp.add(pctr, BorderLayout.CENTER);
cp.add(pright, BorderLayout.EAST);

pack();
show();
}
```

## Example 2, page 4

```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == quit) {
        System.exit(0);
    } else if (e.getSource() == update) {
        tpay.setText(pay());
    } else if (e.getSource() == clear) {
        tpay.setText("");
    }
}
```

## Example 2, page 5

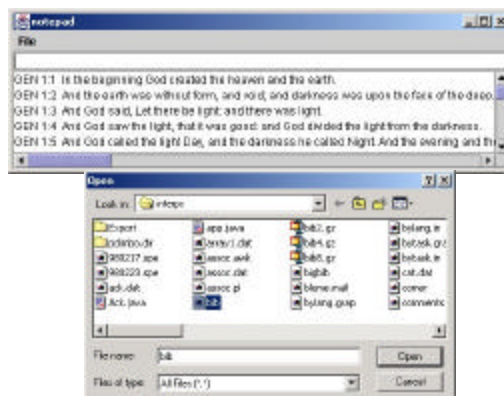
```
String pay() {
    double mp = Double.parseDouble(tpay.getText());
    double prin = Double.parseDouble(tprin.getText());
    double mrate = Double.parseDouble(trate.getText())
        / 12 / 100;
    double totint = 0;
    double totprin = 0;
    String s = "";
    for (int i = 1; i <= 500; i++) {
        double Int = prin * mrate;
        double dp = mp - Int; // decrease of principal
        if (prin - dp > 0) {
            prin -= dp;
        } else {
            dp = prin;
            prin = 0;
        }
        s += " " + i + "\t" + f2(Int) + "\t" +
            f2(dp) + "\t" + f2(prin) + "\n";
        totint += Int;
        totprin += dp;
        if (prin <= 0)
            break;
    }
    s += "\t" + f2(totint) + "\t" + f2(totprin) + "\n";
    return s;
}
```

## Example 2, page 6

```
String f2(double f) { // %.2f, so help me
    NumberFormat form =
        NumberFormat.getInstance();
    form.setMinimumFractionDigits(2);
    form.setMaximumFractionDigits(2);
    //form.setMinimumIntegerDigits(6);
    return form.format(f);
}
```

## Example 3: Notepad editor

- menus
- file dialog
- I/O from file system



### Example 3, page 2

```
public class notepad extends JFrame
    implements ActionListener {
    int last;
    JTextField tf = new JTextField(60);
    JTextArea ta = new JTextArea(15,60);
    JMenuBar mb = new JMenuBar();
    JMenu mfile;

    public static void main(String[] args) {
        notepad a = new notepad();
    }

    notepad() {
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        setTitle("notepad");
    }
}
```

### Example 3, page 3

```
        setJMenuBar(mb);
        mfile = new JMenu("File");
        JMenu tem mi;
        mfile.add(mi = new JMenu tem("Open"));
            mi.addActionListener(this);
        mfile.add(mi = new JMenu tem("Save"));
            mi.addActionListener(this);
        mfile.add(mi = new JMenu tem("Quit"));
            mi.addActionListener(this);
        mb.add(mfile);

        tf.addActionListener(this);

        getContentPane().setLayout(new BorderLayout());
        getContentPane().add(tf, BorderLayout.NORTH);
        JScrollPane jsp = new JScrollPane(ta,
            JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
            JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
        getContentPane().add(jsp, BorderLayout.CENTER);
        pack();
        show();
    }
}
```

### Example 3, page 4

```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == tf) {
        find(tf.getText());
    } else if (e.getSource() instanceof JMenuItem) {
        String b = e.getActionCommand();
        if (b.equals("Quit")) {
            System.exit(0);
        } else if (b.equals("Open")) {
            openfile();
        } else if (b.equals("Save")) {
            savefile(ta.getText());
        }
    }
}

public void find(String pat) { // find next pat in ta
    String s = ta.getText(); // where to search
    if (last + pat.length() >= s.length())
        last = 0;
    int n = s.indexOf(pat, last); // look to end
    if (n == -1) {
        last = 0;
        n = s.indexOf(pat, 0); // look from beginning
    }
    if (n >= 0) {
        ta.setSelectionColor(Color.red);
        ta.getCaret().setSelectionVisible(true);
        ta.select(n, n+pat.length());
        last = n + 1;
    }
}
```

### JFileChooser open file

```
public void openfile() {
    JFileChooser jfc = new JFileChooser();
    jfc.showOpenDialog(this);
    if (jfc.getSelectedFile() == null) // cancelled
        return;

    File fil = jfc.getSelectedFile().getAbsolutePath();
    String f = fil.getAbsolutePath(); // attach
    directory name
    try {
        FileInputStream in =
            new FileInputStream(f);
        byte [] data = new byte [in.available()];
        in.read(data);
        ta.setText(new String(data, 0));
    } catch (IOException e) {
        ta.setText("Can't open file " + f);
    }
}
```

## JFileChooser save file

```
public void savefile(String s) {
    JFileChooser jfc = new JFileChooser();
    jfc.showSaveDialog(this);
    if (jfc.getSelectedFile() == null) // cancelled
        return;

    File fil = jfc.getSelectedFile().getAbsoluteFile();
    String f = fil.getAbsolutePath(); // attach
    directory name
    try {
        FileOutputStream out =
            new FileOutputStream(f);
        out.write(s.getBytes());
        out.close();
    } catch (FileNotFoundException e) {
        System.err.println(e + " can't open " + f);
    } catch (IOException e) {
        System.err.println(e + " savefile error");
    }
}
```

## Applets

- run Java code in browser
- HTML applet tag loads code into a web page

```
<applet
  code="scribble.class" width=500 height=300>
</applet>
```
- default security restrictions on applets
  - can't access client file system
  - can't run processes on client
  - can't create unrestricted top-level windows
  - can't make the interpreter quit
  - etc., etc.
- can use `Socket()` in an applet
  - but can only open socket to system that applet came from
- with Swing plug-in, can do somewhat more
  - but still quite restricted

## Applet scribble (from *Java in a Nutshell*)

```
import java.applet.*;
import java.awt.*;

public class scribble extends Applet {
    int last_x = 0, last_y = 0;

    boolean mouseDown(Event e, int x, int y) {
        last_x = x; last_y = y;
        return true;
    }

    boolean mouseDrag(Event e, int x, int y) {
        Graphics g = getGraphics();
        g.drawLine(last_x, last_y, x, y);
        last_x = x; last_y = y;
        return true;
    }
}
```

## HTML to access an applet

```
<html>
<title> Scribble </title>
<body>
If you're running Java, you should see the
scribble applet below.
<P>
<applet code="scribble.class"
width=800 height=600>
</applet>
</body>
</html>
```

- parameters from applet tag are available to applet

## Standalone scribble (from *Java in a Nutshell*)

```
import java.awt.*;

public class scribble2 extends Frame {
    private int last_x = 0, last_y = 0;

    boolean mouseDown(Event e, int x, int y)
    {
        last_x = x; last_y = y;
        return true;
    }

    boolean mouseDrag(Event e, int x, int y)
    {
        Graphics g = getGraphics();
        g.drawLine(last_x, last_y, x, y);
        last_x = x; last_y = y;
        return true;
    }

    static public void main(String[] args) {
        scribble2 s = new scribble2();
        s.setTitle("Scribble");
        s.resize(600, 400);
        s.show();
    }
}
```

## Socket access in applet

```
public class applci extends Applet {
    String host = "localhost";
    String port = "5194";
    TextField send = new TextField(60);
    TextArea recv = new TextArea(15,60);

    public applci()
    {
        add(send);
        add(recv);
    }

    public boolean action(Event e, Object o)
    {
        if (e.target == send) {
            recv.appendText("cli " + send.getText() + "\n");
            recv.appendText("srv "+echo(send.getText()) +"\n");
            return true;
        }
        return false;
    }
}
```

continued

```
String echo(String s)
{
    try {
        Socket client =
            new Socket(host, Integer.parseInt(port));
        DataInputStream sin =
            new DataInputStream(client.getInputStream());
        DataOutputStream sout =
            new DataOutputStream(client.getOutputStream());

        sout.writeBytes(s); // write to socket
        sout.writeByte('\n');
        sout.flush();
        String r = sin.readLine(); // read reply
        client.close();
        return r;
    } catch (IOException e) {
        return "IOException " + e;
    }
}
```