

Section 7: Applets

1) What is an Applet?

Applets are compiled Java programs that are to be run via appletviewer or a browser. To be run, they must be compiled and invoked from an HTML program using the APPLET tag (see Section 1). By their nature, everything in an Applet is considered graphics.

2) A Minimal Applet

- a) must import the library Applet

```
import java.applet.Applet;
```

- b) must extend the Applet

```
public class stuff extends Applet {
```

- c) example minimal applet:

File "Stuff.java"

```
import java.applet.Applet;  
  
public class stuff extends Applet {  
}
```

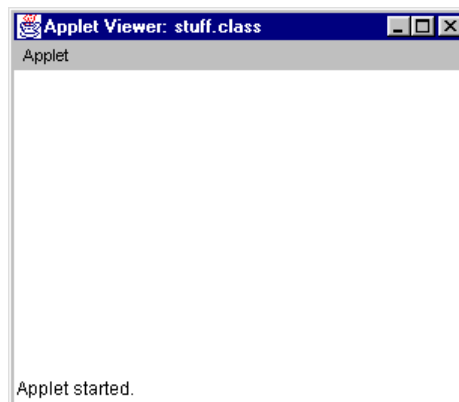
Compiling: `javac Stuff.java // makes Stuff.class`

File "go.html"

```
<applet code="stuff.class" width=300 height=200></applet>
```

Running: `appletviewer go.html`

Result:



3) The Method *paint*

- a) when used, method *paint* is called by Appletviewer and by browsers when an Applet is run
- b) method *paint* must have this header:

```
public void paint(Graphics g)
```

- c) since method *paint* has a parameter of class Graphics, the Abstract Windows Toolkit must be imported

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

- d) example Java program with method *paint*

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

public class stuff extends Applet {

    public void paint(Graphics g){
    }

}
```

4) A program that actually does something!

- a) Class Graphics has many methods (see Section 8). One of them is *drawString()*, which outputs a string starting at the x,y pair indicated. Here is the syntax of a call to *drawString()*.

```
graphics-object.drawString(string, x, y);
```

- b) Example of an applet that actually does something:

file "Greeting.java"

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

public class Greeting extends Applet {
    private x = 50;
    private y = 100;

    public void paint(Graphics g) {
        g.drawString("Howdy, ya'll", x, y);
    }

}
```

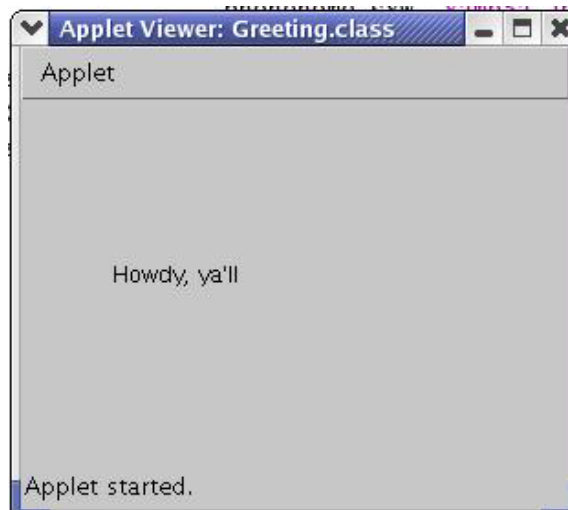
file "example.html"

```
<title>First Java Applet</title>  
<applet code="Greeting.class" width=300 height=200></applet>
```

compiling and running

```
javac Greeting.java  
appletviewer example.html
```

```
}
```



5) The Method `init`

The method `init` has no arguments and, if used, is executed first (before `paint` is called). Here is an example:

```
import java.awt.*;  
import java.applet.Applet;  
  
public class Greeting extends Applet {  
    private x;  
    private y;  
  
    public void init() {  
        x = 50;  
        y = 100;  
    }  
  
    public void paint(Graphics g) {  
        g.drawString("Howdy, ya'll", x, y);  
    }  
}
```

6) Constructors

- a) Constructors are functions that have the same name as the class.
- b) Constructors are always public and have not return type.
- c) Constructors can have no, 1, 2 or as many parameters as desired.
- d) Used in place of method init when a class is used by another class.
- e) Example:

```
import java.awt.*;
import java.awt.event.*;

public class Example1 {

    private int x, y, w, h;
    private String s;

    public Example1(String initS, initX, initY) {
        s = initS;
        x = initX;
        y = initY;
    }

    public void paint(Graphics g) {
        g.drawString(s,x,y);
    }

}

import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;

public class Example2 extends Applet {

    private Example1 a = new Example1("spam", 10,10);
    private Example1 b = new Example1("catsup",100,50);

    public void paint(Graphics g) {
        a.paint(g);
        b.paint(g);
    }

}
```