Applets

getting started

A Java **applet** is a small application written in Java which runs within the context of a larger program. Primarily, we will be interested in applets that run within the context of a web browser. This is accomplished by embedding a Java class file into an `html` webpage.

The applet is automatically downloaded from the server which is hosting the webpage and it is run on the client machine. This allows for interactive and dynamic webpages that don't put any strain on the server, but off-load this work to the client. Of course, this creates security vulnerabilities, as the server could provide malicious code to the client. Hence, all applets are run in a *sandbox* – i.e. a part of the client machine which has limited access to system resources and the filesystem.

Though applets are not terribly pervasive technology nowadays, they serve as an effective tool for learning about web-based programming and graphical user interfaces (GUI). The concepts acquired while working with applets will be easily transferrable to other web-based and GUI programming.

`HelloWorld`

The minimal conditions for writing an applet are:

1. Your class must import the *Abstract Window Toolkit*
2. Your class must import the `Applet` class
3. Your class must be a `public` subclass of the `Applet` class

The *Abstract Window Toolkit* (AWT) is a framework provided by Java for creating interactive graphical user interfaces. It may be imported with the following statement:

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

However, `java.awt` is one of the largest packages in the Java library so, it is may sometimes be wise to simply import the classes which we need. Another package that will need to be imported is that which contains the `Applet` class:

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

`java.applet` is a much smaller package than `java.awt` and contains the basic necessities for creating an applet including the `Applet` class. As mentioned previously, all applets that are created must be subclasses of `Applet`. This is becuase the `Applet` class takes care of setting up lots of environmental information. Everything that is predefined in the `Applet` class takes care of all communication with the browser so that the programmer does not have to. The programmer simply inherits all the functionality of the `Applet` class and then overrides the methods they need to customize the applet.

The code below is the skeleton of an applet. In fact, the code will even compile and run, it simply won't do anything.

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

public class HelloWorld extends Applet { }
```

Of course, we would like our applet to actually do something. So, we override methods that are inherited from the `Applet` class. There are a number of relevant methods that play a role in the process of building an applet, but for now we will simply be concerned with a method

called `paint()`. This method is somewhat analagous to the `main()` method in a command line application, though not quite the same.

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

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        // code here
    }
}
```

Notice that the given parameter is a `Graphics` object called `g`. This object represents a graphic context and it will be our means of drawing on our applet. Some of the information encapsulated in this `Graphics` object is: font, color, etc. The most important piece of information in the `Graphics` object is the thing which is being drawn on: in this case, our applet.

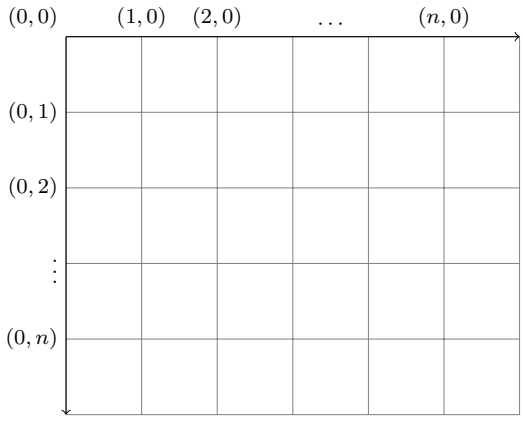
The `Graphics` class provides lots of drawing methods, one of which is: `drawString()`. This will be the key to our `HelloWorld` program.

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

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello, World!", 20, 20);
    }
}
```

There are a few things to notice. We do not print `String`s in applets. The applet is not text based, it is graphical and so everything is represented as an image. Therefore, we do not simply provide a `String` which we want to draw, but also the coordinates for a starting location.

The coordinate system of Java's graphical framework begins in the upper left hand corner of the relevant component (in this case the applet). The coordinates increase downward and to the right. This may be thought of as the fourth quadrant of the Cartesian plane.

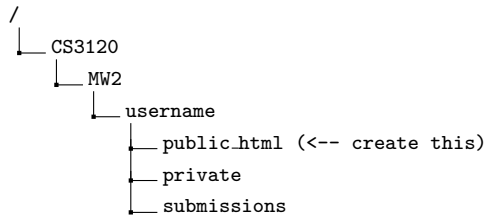


In the case of `drawString()` above, the bottom left corner of the text begins at the coordinate (20, 20). In other words, imagine a horizontal line drawn through the point (20, 20). The text would be written immediately above that line.

public_html

We now have our completed `HelloWorld` applet which we can compile just as we would any other Java program. However, we do not yet have a means of running it, as applets are meant to be executed within the context of a web browser. In order to run this applet in a web browser, you will have to embed the code in a webpage.

First, you will have to create a directory in your home directory called `public_html`. The directory structure should look like this:



To create the directory, simply type the following command from any directory:

```
mkdir $HOME/public_html
```

The first part of the path is an environment variable which is different for each user. It signifies the users home directory and replacement is done automatically by the shell. In the case above the shell would replace `$HOME` with:

```
/CS3120/MW2/username
```

Your `public_html` directory will serve as the container for all files which you would like to make accessible publicly through the internet. Anything in this directory (with the proper permissions) can be accessed through a web browser.

When you first create the directory, it will, of course, be empty. So, there will be nothing to access. The first thing you must do is create a file in `public_html` called: `index.html`.

In `index.html`, you will minimally need to write the following:

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

The line above is called an applet tag. The applet tag lets the web browser know that an applet is embedded into this webpage and that it should be allotted a space on the page of 200 by 200 pixels (specified by the `width` and `height` attributes). The `code` attribute specifies where to find the applet which should be executed on the client machine.

There are a few other attributes that can be used with the applet tag which we may discuss at a later point. A description can be found here.

To see your applet through a browser, all you need to do now is visit your personal webpage. Your personal URL will take the following form:

```
http://notnotbc.org/~username/
```

This will automatically read the file: `index.html`. Then, you should be prompted by the

browser about whether you trust the applet. If you proceed, you will finally see your applet being executed. You will definitely need to install Java on your personal machine and may have to white list the class server. At any rate, the following is a link to a working `HelloWorld` applet:

`http://notnotbc.org/~michael.g/`

Use this as a guide to write your `HelloWorld` applet. You may simply copy and paste all of the code and html – *the point is to get things working!* If you have any issues, please do let me know.