Applet  a specially defined Java class that extends the behavior of a web browser in a user-specified way, creating a user-programmable area within a web page. Applets allow new functionality to be delivered to users over the web. As a precaution, applets operate in a security sandbox which limits their functionality in certain ways meant to protect web users.

array  a group of like (same type) data elements which share a name—such as coordinate points in a drawing, colors in a palette, primitive drawing objects in a graphics application, and so on. The individual elements of the array are accessed using an index (an integer) counter to point to the particular element. The index starts at zero, so the third member of a list is myList[2].

    int[] myIntegers = new int[10];  // an array with space for 10 integers
    int[] moreInts = {0, 4, 7, 4};  // an array containing 4 integers
    Point[] points = new Point[4];  // an array to hold up to 4 “Points”

ArrayList  A more complex, but flexible form of array, array lists may have elements added or removed at will, unlike simple arrays. Included in this course as a general example of more sophisticated collections.

assignment  transferring the contents of one memory location to that of another, as when writing a=25+22 which assigns the value computed on the right of the equal sign to the storage space identified on the left side of the equal sign. NOT a statement of or a test for equivalence.

AWT  The “first generation” abstract window toolkit Java package for graphics and interactive user interface creation. AWT uses heavyweight components for display, in contrast to Swing’s lightweight components.

boolean  (boolean) data that resolves to values true and false. Often used as part of loop and branch constructions.

branch  any of several constructs (if, switch, etc.) that perform some sort of a test to decide if an instruction or instruction block should be executed.

bytecode  the compiled form of a Java applet or application that is ready for the Java Virtual Machine (JVM). These are the “.class” files transferred to the web server.

casting  the operation of converting one type of data to another. Not all types may be interchanged, but where they can, casting provides a “legal mechanism” for doing so (otherwise Java’s ‘strongly typed’ character would prohibit such ‘mistakes’). Indicated by enclosing the resultant data type in parentheses:

    int oddInt = (int) 24.0/7.0; // casting division results to int

Char  (char) a single character in Java (can be any Unicode character in addition to ASCII characters). Represented in code surrounded by single quotes (‘a’), in contrast to Strings.

code block  a section of Java code, enclosed in curly braces {like this}. Anywhere a single instruction can be used, a code block can be used (as part of a loop, branch, method definition, etc.).

comment  Text in the Java source code that doesn’t influence program execution. Comments are provided for future human readers of the source. Inline comments start with two slashes “/" like this” and end at the end of the source code line. Block comments begin with “/*” and end with “*/”. All text between the beginning and end is comment text.

constructor  The method within a class which is responsible for processing a new object request. Assigns values to class fields, etc.

data  information, encoded for manipulation by the computer. May include “primitive” types such as numbers (integers, floats, doubles), text (chars, strings), as well as “higher level” types such as Colors, coordinate points, and mouse events.

Double  (double) like a float, but using 64-bits, so more precise and also a larger range of values.
event handler a method which responds to a particular class of events. mouseClicked is one of the event-handlers expected by the MouseListener interface. In addition to adding a mouseClicked event to your applet, you must include

```java
addMouseListener(this);
```

in the Applet’s init() method, and add “implements MouseListener” to the Applet header (class declaration).

exception an error condition that prevents the “normal” continuation of the program. Possibilities include runtime division by zero (which is undefined), user input of ill-formed data (“123t3” as a number?), etc. Exceptions are said to be thrown when they first occur. They may be caught by program code and handled internally, or allowed to propagate through the application, usually causing an error message and possibly program termination.

extend fundamental concept in OOP, used when defining a new class that has all the features of the old class, plus some more. Features (methods, fields) borrowed from the parent class do not need to be implemented in the child. Inheritance is automatic.

field an element of data associated with a class (aka object). These values are part of the object definition, not transient data calculated as part of some method execution.

Float (float) one of several numerical representations that use the idea of scientific notation. Like int, it uses 32-bits, but allocates them differently, expressing a number as several significant digits and an exponent, as in 3.1415976(104). These numbers are necessarily approximate, but can represent numbers like ¼, π, and e that lie between integers, or are very large (astronomic distances) or small (molecular distances).

Graphics a Java data object used to handle 2D graphic output. These objects are passed to applets from browsers using the applet’s paint method allowing the applet to create output. In addition to the drawing pixels themselves, a Graphics has a current color and current font. It does not have a separate 2D coordinate system or line-style (stroke) attribute, which did not appear until the Graphic2D object was defined.

Graphics2D an extension of a simple Graphics, the Graphics2D adds a separate coordinate system and all affine transformations (scaling, rotation, translation) for all output, allowing rotated text and rectangles. It also incorporates line-style and line-weight concepts (known as stroke data).

GUI Graphical User Interface—the general name for a computer interface that uses graphical output and mouse input to control what the application does. Almost all modern programs and operating systems use GUIs.

heavyweight Used to describe Java methods that use high-level routines (called peers) in the host operating system to display buttons, sliders, dialog boxes, etc. This strategy produces an interface on the Java application that looks like those of the host operating system, but has proven hard to support. It has largely been supplanted by use of lightweight (Swing) components.

IDE integrated development environment, applications like eclipse that help compose and compile, document and debug application code. They provide help keeping the code organized, renaming variables and methods, displaying context-specific options, suggesting corrections, etc.

implements a Java keyword indicating that a class provides the methods necessary for a particular interface. Commonly added to an Applet to include Mouse input, as

```java
public void myApplet implements MouseListener, MouseMotionListener ...
```

which “promises” that five specific methods (event handlers) will be implemented in the Applet: mouseMoved, mouseDragged, mouseEntered, mouseExited,
mousePressed, mouseReleased, and mouseClicked. Some (most?) implementations may be empty, but they must exist.

instance an object or class as used in an application. The class source code defines the data and methods of the object as a template, but don't actually make one. The new operator creates an instance of the class. An application may create multiple instances of a class (think “bouncing balls”) in some cases.

Integer (int) positive or negative whole number (…,-2,-1,0,1,2,3,…). The number must fit into 32-bits of memory. The JVM has a maximum (and minimum) definable integer. Attempts to create a bigger or smaller number cause an error condition, or exception to be thrown. These representations are exact.

interface a set of methods required to completely implement a general feature. We used (or at least looked at) the following interfaces: “Runnable”, “MouseListener”, “MouseMotionListener”, “ItemListener” and “ActionListener”. See implements and listener for more.

Java a computer programming language created by Sun Microsystems and incorporating concepts of a “java virtual machine” [JVM] and “object oriented programming” [OOP]. Java is current at version 1.6, though our projects were intended to use only version 1.5 or earlier (in order to expand the number of machines which would have a suitable JRE).

JRE “Java Runtime Environment” – The particular configuration of Java available on a specific host. Since Java must be downloaded and installed on a computer before it is available to use, and software is updated or revised from time to time, it is possible that the JRE on a particular machine is not that expected by the bytecode of a particular applet, in which case the applet will fail to run. You can usually run a 1.4 applet on a 1.6 JRE, but not the other way around.

JVM “Java Virtual Machine” Each hardware CPU has a unique set of machine-level instructions, limiting the ability to run software from a different computer. A virtual machine is a piece of software that acts like a hardware CPU, but translates or interprets instructions for the actual CPU on which it runs. A standardized java virtual machine means that software can be created that runs “the same” on many different computers (Mac, PC, unix, etc.).

Layout A “layout” is the general strategy for positioning components (buttons, panels, etc.) within the application window without taking exact-location responsibility. Several generic layout strategies and related layout managers have been identified: including flow, border, grid, and grid-bag.

life-cycle applets are connected to their browser environment through a set of pre-defined methods which the browser expects to find available from the applet: init(), start(), paint(), stop(), and destroy(). When the applet is “run” by the browser, the browser calls these methods in the order shown. The applet author must take responsibility for defining appropriate actions for the methods they choose to implement.

lightweight Used to describe Java methods that construct the graphic display of buttons, sliders, dialog boxes, etc. entirely within Java This strategy produces an interface on the Java application that looks the same across different host operating systems. It has largely supplanted by use of heavyweight (AWT) components.

listener As part of making a component (class) responsible for handling some type of event, the application generally adds an event listener to the class. Doing this tells the JRE to notify your event handler when an event of that type occurs.

loop any of several constructs (for, while, etc.) that create instructions that are executed multiple times. Makes code more compact when doing things repeatedly.

method a named block of code, optionally having parameters or arguments, that performs some operation. Some methods are built-in to the language (Math.round(), etc.), others are created by users.

new the Java operator that creates an instance of a class.
OOP  
“Object Oriented Programming” A computer-programming strategy in which programs are broken up into blocks of code which completely encapsulate both behavior (program action) and information (data description) of an aspect of the program. A class is the mechanism for defining objects in Java.

public  
a keyword used when defining classes, methods, and fields that sets the scope of their use. Public fields may be referenced from elsewhere in the application without using a method. Public methods may be used from other packages or classes, and public objects from classes may be instantiated anywhere within the application. The goal is to encapsulate (think “hide”) the inner workings of a method or class and prevent unintended results (“side effects”) of code changes.

sandbox  
The Java environment may be used to create full-function applications as well as web-based applets. To prevent applets which come over the web from having the same access to local computing resources (user disk, etc) as a local application, a security scheme was created to limit access when running as an applet. Applets are thus said to run in a sandbox with limited functionality.

For applets, the sandbox dictates that all resources (files) retrieved by the applet must come from the same server as the applet.

Scope  
the portion of the program within which a particular variable has meaning. Generally, the scope of a Java variable is the block within which it is defined.

Sleep  
When a thread stops executing for a preset period of time it is said to sleep. During this time it is not using CPU resources, which are allowed to flow to other threads on the system, so putting a thread to sleep is a very efficient way of inserting a pause. However, the thread is unable to do anything during the time it is asleep, one reason not to sleep an application’s main thread.

String  
A group of characters in Java. Represented in code surrounded by double quotes (“abc”) in contrast to chars.

Swing  
The “second generation” Java package for creating and managing Graphical User Interfaces. Methods are identifiable by a “J” prefix, as in JApplet, and JButton. Swing uses lightweight components, in contrast to AWT, which uses heavyweight components.

this  
used to self-reference a class when interacting with other parts of the application. Particularly useful if method parameters have the same names as internal field parameters, as illustrated here:

```java
public class P2D {
    double x,y;
    public P2D ( double x, double y ){
        this.x = x;
        this.y = y;
    }
}
```

thread  
An identifiable sequence of instructions or steps that the program executes is a thread. Modern operating systems allow multiple threads to exist simultaneously. In Java, threads are used to separate execution of sequences that might normally interfere with each other (pausing and waiting for user input while counting down a timer, for example). Threads are very helpful for animation and other “over time” procedures.

try-catch  
a language construct which identifies a block of code (the try block) within which exceptions will be routed to the one or more catch blocks.

variable  
the concept of a named memory location within which a piece of data may be stored.

void  
used in method declarations (e.g. “public void paint( Graphics g)” ) to indicate that the method does not return anything. The alternative to returning nothing is to return a primitive data type (e.g. “public float average( float[] vals);”) or higher-order objects (e.g. “public Graphics getGraphics()”).
Be sure you can read the applet below and understand everything in it. This is a very simple example and should not be challenging. The exam may well include a sample block of code like this. I may ask you to identify and correct common syntax errors in the code, or sketch out the output that would result from running the code.

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

public class myWorld2 extends Applet {

    // We don't need "init" or "start" because we're such a simple applet.
    public void paint( Graphics g )
    {
        Font f1 = new Font("Serif",Font.PLAIN, 72); // define a Font setting
        g.setFont( f1 );                              // apply the Font setting
        dropShadow( g, "Beautiful Day!",8,300 );      // draw some drop-shadow text
        dropShadow( g, "Hello World!",8,198 );        // more drop-shadow text
    }

    // Define a method for doing drop-shadow text.
    private void dropShadow( Graphics g, String text, int x, int y ){
        Color[] colors = {Color.lightGray,Color.gray,Color.black};
        System.out.println("Printing out: "+text );  // Write text to console for debugging
        for( int i=0; i<colors.length;i++ ){
            g.setColor( colors[i] );                  // setting the color
            g.drawString( text, x-i*2, y-i*2);        // and drawing the text
        }
    }
}
```
Sample Questions

1. I want to add an interface to my applet that has buttons that look the same on all computers. Do I use AWT or Swing?

2. Applications do not run “in the sandbox”. Give an example of something an application can do that an applet cannot?

3. Java is an “object oriented” programming language. What is an “object”? Does “object” equate with “class”, “method”, “package”, or “application”?

4. What is the difference between the “int” and “float” data types?

5. Give an example of two or more “flow control” (branching) statements in Java.

6. What is the purpose of a Layout Manager? Name two of them.

7. Java is said to be “object oriented”. What is an object? Does object correspond to class, method, package, applet, or application?

8. What five methods might constitute the “life-cycle” of an applet? Why is the term “life-cycle” appropriate?

9. If I have an applet that draws a line and I want to make my applet use mouse input to set the endpoints of the line, what three generic changes will I need to make to the applet?

10. Write an applet to display “Hello World” in the browser.

11. I have an applet that plays a tone (I have several) when I click the mouse. I’d like to make a web page that plays different tones when I click different areas of the page. Can I do this somehow with just one applet, or do I need to make a different applet to play each tone?

12. When might I use a “thread”?

13. If variable “colors” is an array of Colors, write the statement that will set the second color to red.

14. Identify two differences between a Graphics and a Graphics2D.

15. What does double buffering refer to?

16. There seem to be two nearly identical classes for many user-interface objects, like Button and JButton. What’s the “J” indicate?