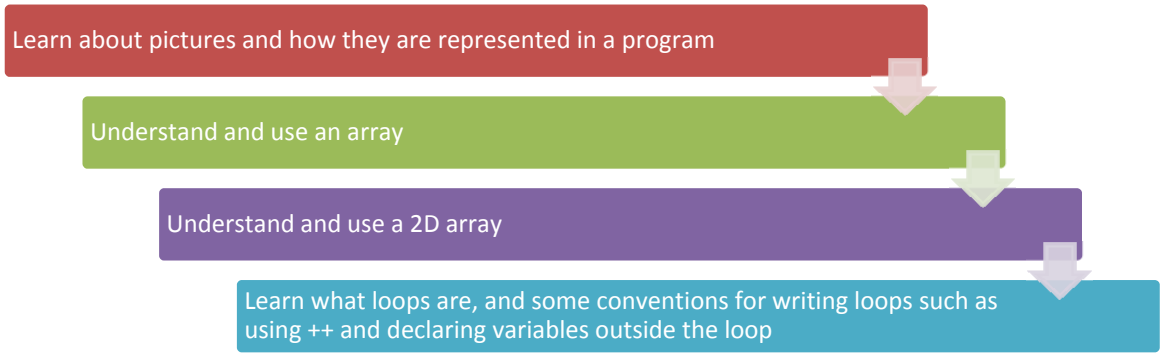
Module II: Summary

Learning Objectives



Programming concepts

- Programs, as you have seen before, are made up of a series of statements that execute in order. Each of these statements can contain *expressions* which are evaluated in a certain order to produce final values that are substituted in to methods. Examples of expressions are 3 + 7 and "str" + "ing". Some expressions like "string" and 2 have their value already fixed when you write the code: these are called *literals*
- An *array* is a way of storing a list of similar items in one continuous chunk of memory. We can store and retrieve these values using an index with *pixelArray[index]*. We declare an array like this Pixel[] pixelArray = pictureObj.getPixels(); . An array may hold objects or primitive types (as long as everything in the array is of the same type). Arrays are indexed from 0, so an array of size array.length will have indexes from 0 to array.length - 1
- *Classes* in Java are organized into *packages*, which we need to import to tell Java we want to use them. For example, we can use import java.awt.*; to import all the classes in java.awt, or we can use import java.awt.Color to import just the Color class. If you forget to import a class, but try to use its name in your program, it will not compile and you will get an undefined class error. When you create a class in the same directory and compile it, Java will automatically pick it up for you so you don't have to worry about including packages.
- A two-dimensional array is an array that can be indexed in two dimensions. It is essentially an array of arrays, where each of the subarrays is the same length. We can define a 2-D array with something like int[][] int2d = new int[4][5]; which in this example defines a 4 columns by 5 rows int array. We access and modify a value with int2d[col][row].
- You can use a while loop to repeat the same thing until a condition is true. For example:

```
int i = 1, a = 1;
while (i <= 5) {

        a = a * i;
        i++;
}
```

  The above simple loop uses a while loop to find the product of the numbers from 1 to 5.
- Java gives us some useful shortcuts for modifying the values of variables based on the existing variable. x += a adds a to x, x -= a subtracts a from x. x *= a multiplies x by a and x /= a divides x by a, where a is any expression. There are also the increment (x++) and decrement (x--) operators, to add and subtract 1 from a variable, which are very commonly used.
- You can declare variables anywhere you want, but when you declare variables inside a set of braces, they are only valid within that set of braces. It is usually a good idea to declare variables in the smallest scope (the closest set of braces) possible, to avoid bugs and name collisions, and to make it easier for the Java compiler to optimize your code.
- A for loop is a way of taking common operations you might do in a loop and giving them a simpler syntax. We can write the code we wrote above in a while loop as a for loop by doing:

```
int a = 1;
for (int i = 1; i <= 5; i++) {
    a = a * i;
```

```
}
System.out.println("5*4*3*2*1 = " + a);
```

With a for loop, we can put the initialization and incrementing code that we commonly use in the same place as our loop test, so it is easier to understand what our loop is doing. This also helps to ensure that we don't forget these important steps in the loop.

- For certain types of collections and arrays in Java, we can also use a for-each loop, which is like a for loop except it assigns each object to a variable, and doesn't make you worry about how each element is accessed (e.g. by tracking the index). A for-each loop is written like:

```
for (Pixel pix : pic.getPixels()) {
        // do something with each Pixel
}
```

Media concepts

- You should know how to create a Picture object and display that picture on the screen.

```
String file = FileChooser.pickAFile();
Picture pic = new Picture(file);
pic.show();
```

- A *pixel* is a the most basic unit of digital images. A pixel represents a single block in the image as it is displayed on your screen, which can be assigned a color by adjusting the three primary color values (red, green and blue)
- When we manipulate an image in Java, it is taken from its normal representation on disk to a format which is easier to manipulate programmatically. Usually on disk the image is stored compressed in either a lossless compression format (like PNG or GIF) or a lossy compression format (like JPEG). Occasionally images are also stored in bitmap (BMP) format, which has no compression at all and takes up a lot more space since we have to store 3 bytes for each pixel.
- When we read images into Java, they are usually stored in an array. In this representation, we might access the 100th pixel with Pixel p = pixelArray[100].
- We can represent a pixel in our image with a Pixel object, which has a color (RGB values) and a location.
- Another way of thinking of an image is as a two-dimensional array of pixels. Unlike a traditional 2D Array, the pixel array is accessed using pictureObj.getPixel(x, y). In this case, x refers to the horizontal position (i.e., column) of the pixel, and y refers to the vertical position (i.e., row) of the pixel.
- We can write algorithms that do things to pictures. An example we presented in this module is changing the color of all of the pixels in a picture (for example, decreasing the amount of red).