An array is very useful for tabulating a *frequency distribution*, which is a table showing the number of times a statistic is distributed across a range of values.

Let’s pretend that for a science project, Jill must compile a frequency distribution showing the heights of plants that she has grown for the project.

First, she takes a sheet of paper, draws a table of blocks on it and labels each with a range of heights.

Next, she measures each of her plants and places a tally mark in the box corresponding to its height.

When finished, the number of marks in each box shows how many plants are in that height range. For instance, the figure at right indicates that 3 plants were 0 to 9” tall and 5 were 10 to 19” tall.

Let’s write a Java application that Jill can use to compile her frequency distribution.

The program must compile the table so that Jill need only enter the heights of her plants. The algorithm follows the process Jill would use if she were to create the table by hand.

A five-element array `height` with component data type `int` serves as the table. We can build and initialize it with this statement:
int [] height = { 0, 0, 0, 0, 0 };

Jill will input the plants’ heights using a dialog box. She’ll cancel the dialog box when she’s done entering input. The code outline for the loop is:

```
while ((input=showInputDialog( prompt )) != null )
{
    hgt = Integer.parseInt( input );
    . . .
}
```

`prompt` is a `String` object holding the user prompt and `input` is a `String` object used to capture Jill’s input. If she cancels the input dialog, `showInputDialog` returns `null` and the loop quits.

For each height, we must increment the array element corresponding to that height:

```
position 0 for heights 0 to 9
position 1 for heights 10 to 19
position 2 for heights 20 to 29
etc.
```

We can calculate the position from the height by dividing the latter by 10. This integer division knocks the ones digit off the height. For example, 9/10 gives 0, 16/10 gives 1, 22/10 gives 2, and so forth. The Java statement is:

```
height[hgt/10]++;
```

Heights of 40” and above must be dealt with using an `if` statement so that the program doesn’t give an `ArrayIndexOutOfBoundsException` run-time error.

The entire application is shown on the next page.
import static javax.swing.JOptionPane.*;

public class JillsFreqTab
{
    public static void main( String args [ ] )
    {
        // declare data
        String prompt = "Enter plant height:";
        String input; // input string
        int hgt;       // input height
        // build frequency table
        int [ ] height = { 0, 0, 0, 0, 0 }; // if height has first digit 0-3
        // loop until user cancels dialog
        while ( (input=showInputDialog( prompt )) != null )
        {
            hgt = Integer.parseInt( input );
            if ( hgt < 40 ) // if height has first digit 0-3
                height[hgt/10]++; // increment corresponding position
            else
                height[4]++; // else increment last position
        }
        // print results
        String output = " HEIGHT   COUNT" + "\n"
        + "00 to 09" " + height[0] + "\n"
        + "10 to 19" " + height[1] + "\n"
        + "20 to 29" " + height[2] + "\n"
        + "30 to 39" " + height[3] + "\n"
        + "over 39" " + height[4];
        showMessageDialog( null, output );
    }
}
Programming Exercise

1. Write a Java application or applet that you and your friends can use to vote for your favorite teacher, music group, ice cream or whatever.

   Your program must display a ballot similar to that shown on the right. The user enters his or her vote by typing 1, 2, 3, etc. and clicking OK. Your program must record the vote by incrementing an array element corresponding to each selection.

   You must check that the user types a valid selection. For example, if he or she enters a 5, your program must respond with an error dialog such as that shown here.

   Your program must use a loop to repeatedly present the ballot and record the user's vote. Quit the loop when the user clicks Cancel or Close.

   When the loop terminates, display a dialog box with the results of the voting.