

# Lab Exercise for Assignment 6

## Intro to Programming: Lab Exercise for Assignment 6

---

- Assignment due Dec 6 14:30

### Goals

The exercise gives you practice at writing GUI programs and programs using ArrayLists.

- Copy the files in Files for Assig6 to your USB or download the [Zip file](#).

### GUI Programs (Do in the first week)

---

#### Circles

`CirclesExercise` lets the user place small and large circles on the graphics pane using the mouse. It needs one field to store the "current size". It has three buttons: **Small**, **Large**, and **Clear**.

- The **Small** button should change the "current size" to 10,
- The **Large** button should change the current size to 30.
- The **Clear** button should clear the graphics pane.

When the user releases the mouse at any point on the graphics pane, the program should draw a blue circle of the current size at that point.

Note that the program should not draw anything when the **Small** or **Large** buttons are pressed - it should just change the value stored in the field.

You need to

- declare a field to store the current size. Initialise it with "Small"
- define a constructor which sets up the GUI:
  - sets the mouse listener,
  - adds the three buttons.
- define methods to respond to the **Large** and **Small** buttons by setting the current size field to an appropriate value.
- define the `doMouse` method which will respond when the user releases the mouse at a point by drawing a blue circle of the current size at the point.

---

#### Words

`WordsExercise` lets the user place words on the graphics pane using the mouse. It needs one field to store the current word. It has two buttons and a text field:

- The **Clear** button should clear the graphics pane.
- The **Set Word** textfield should let the user enter the current word
- The **Color** button should set the color of the UI to the color chosen by the user.

When the user releases the mouse at any point on the graphics pane, the program should draw the current word at that point.

Note that the program should not draw anything when the user enters a word - it should just change the value stored in the field.

The field definition and the constructor are already written. You need to

- define methods to respond to the **Set Word** textfield and the **Color** button.
- define the `doMouse` method which will respond when the user releases the mouse at a point by drawing the value of the word field at the point.

**Note:** you can use the `JColorChooser.showDialog(null, "Choose Color", null)` method to ask the user for a new color.

---

## Stars

`StarExercise` lets the user draw "stars" on the graphics pane. Stars consist of a set of lines all drawn from a center. The program has two "modes":

- setting the center of the star,
- setting the points on the star.

When the user clicks the **Set Center** button, the program changes the mode to setting the center

**Note:** clicking the button does not draw anything!

When the user clicks the "Set Points" button, the program sets the mode to setting points

**Note:** clicking the button does not draw anything!

When the user clicks the mouse:

- If the mode is setting the center, it draws a little circle of size 4 at the position the mouse was released, and stores the position as the current center
- If the mode is setting the points: the program draws a line from the the current center to the position where the mouse was released

The user can click the **Clear** button to clear the graphics window.

You need to

- declare fields to store the mode and the current center point
- define a constructor which sets up the GUI: sets the mouse listener, and adds the three buttons.
- define methods to respond to the **Set Center**, and **Set Points**, buttons.
- define a `doMouse` method which will respond when the user clicks the mouse at a point. It must consider the state to determine whether to change the center point or to draw a line.

---

## Programs with ArrayLists (Do in the second week)

Each of the following methods is relevant to the main assignment program in a different way. Note: Several of the methods involve reading values from a file. You can use the **Show file** button to display the contents of a file.

### (a) `plotNumbers()`

Reads a sequence of numbers from a file into an `ArrayList`, and then plots the numbers as rectangles. The method should create an `ArrayList`, then read numbers from the file `numbers.txt`, adding them to the end of the `ArrayList`.

It then iterates through the `ArrayList`, plotting each number as rectangle. The number specifies the height of a rectangle; the width of the rectangle is 5. Test on `numbers1.txt` and `=numbers2.txt`

### (b) `readAndReverse()`

Asks the user for a file and reads the tokens from the file storing them in an `ArrayList` of `Strings`. (It uses an `ArrayList`, not an array, because it does not know how many tokens there are in the file until it has read them all.) It then prints out the number of tokens it read and then prints the tokens in reverse order, one per line. Test your method on the files `a-story.txt` and `b-story.txt`.

### (c) `plotColoredNumbers()`

Reads and plots numbers, just like `plotNumbers` except that the rectangles are colored.

After reading the numbers into an `ArrayList`, the method finds the number half way through the sequence of numbers. (If there are *count* numbers, then the half-way number is at index  $(count-1)/2$  )

It then plots each number as in `plotNumbers`, but the rectangle should be green if the number is less than the half-way number, black if it is the same as the half-way number, and red if it is more than the half-way number. Test on `numbers1.txt` and `=numbers2.txt`

### (d) `printDifferences()`

When analysing sequences of numbers (eg from a digital signal), it can be useful to compute the difference between each pair of numbers in the sequence. (The differences are an approximation of the derivative).

The `printDifferences` method reads numbers from a file into an `ArrayList`, and then calculates and prints the difference of each pair. Test on `numbers1.txt` and `numbers2.txt`

For example, if the numbers were: 4, 7, 12, 8, 15, 27 it should print: = 3 5 -4 7 12=

---