
Data Structures and Algorithms

COMP 103

2019-20

Semester 2

Lecture 02c

Dr. Kerese Manueli

kerese.manueli@ecs.vuw.ac.nz

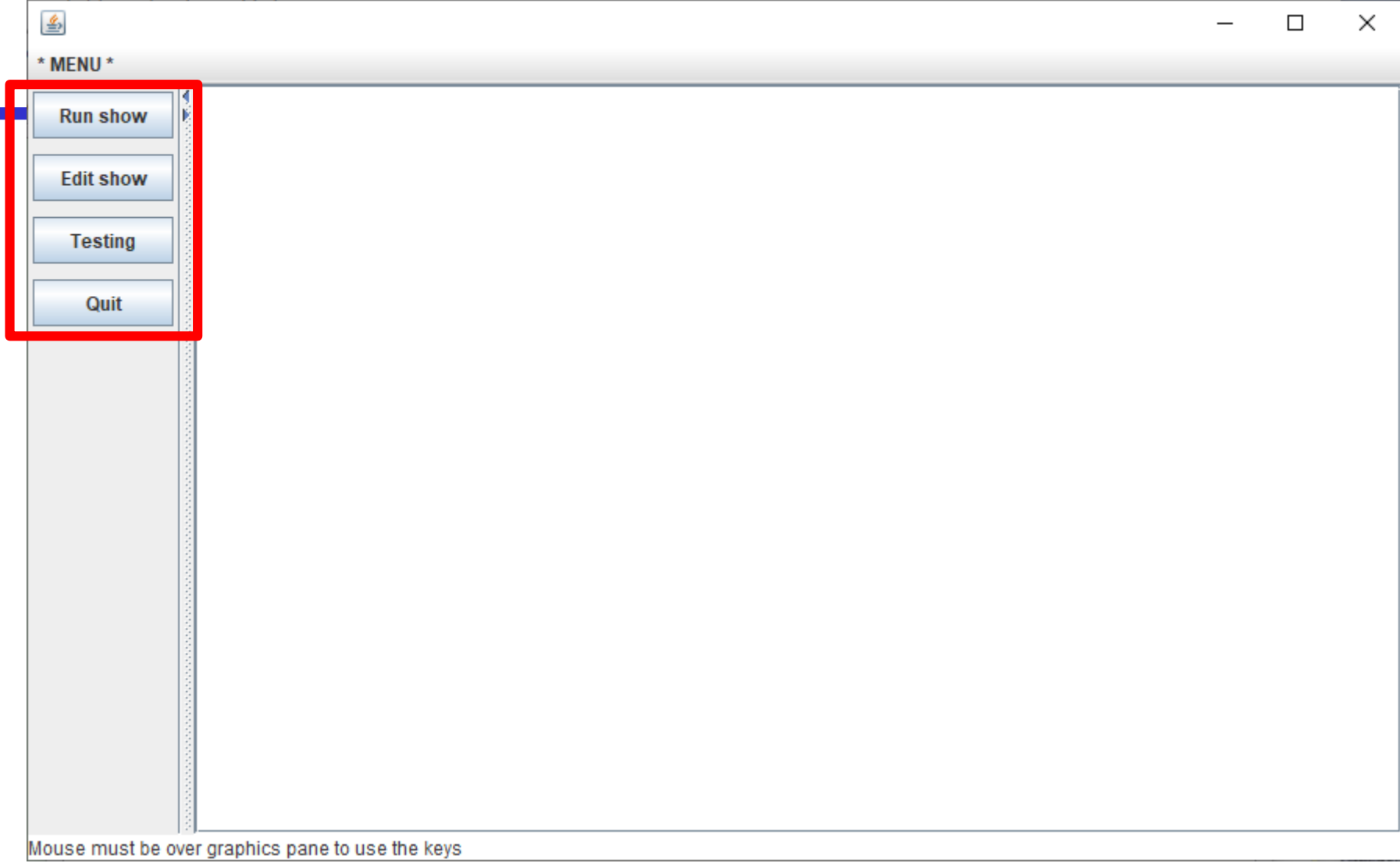
Victoria University of Wellington

```
SlideShow - SlideShow
Class Edit Tools Options
SlideShow x
Compile Undo Cut Copy Paste Find... Close Source Code
38 /**
39  * Constructor
40  */
41 public SlideShow() {
42     /*# YOUR CODE HERE */
43     images = new ArrayList<String>();
44     setupGUI();
45 }
46
47 /**
48  * Initialises the UI window, and sets up the buttons.
49  */
50 public void setupGUI() {
51     UI.initialise();
52
53     UI.addButton("Run show", this::runShow);
54     UI.addButton("Edit show", this::editShow);
```

Angle brackets < and >
Round brackets (and)
Curly brackets { and }

```
SlideShow - SlideShow
Class Edit Tools Options
SlideShow X
Compile Undo Cut Copy Paste Find... Close Source Code
52
53 UI.addButton("Run show", this::runShow);
54 UI.addButton("Edit show", this::editShow);
55 /* UI.addButton("add before", this::addBefore);
56 UI.addButton("add after", this::addAfter);
57 UI.addButton("move left", this::moveLeft);
58 UI.addButton("move right", this::moveRight);
59 UI.addButton("move to start", this::moveStart);
60 UI.addButton("move to end", this::moveEnd);
61 UI.addButton("remove", this::remove);
62 UI.addButton("remove all", this::removeAll);
63 UI.addButton("reverse", this::reverse);
64 UI.addButton("shuffle", this::shuffle); */
65 UI.addButton("Testing", this::setTestList);
66 UI.addButton("Quit", UI::quit);
```

Relocate the * and /
to the end of code
line 64



```
SlideShow - SlideShow
Class Edit Tools Options
SlideShow X
Compile Undo Cut Copy Paste Find... Close Source Code
52
53 UI.addButton("Run show", this::runShow);
54 UI.addButton("Edit show", this::editShow);
55 /* UI.addButton("add before", this::addBefore);
56 UI.addButton("add after", this::addAfter);
57 UI.addButton("move left", this::moveLeft);
58 UI.addButton("move right", this::moveRight);
59 UI.addButton("move to start", this::moveStart);
60 UI.addButton("move to end", this::moveEnd);
61 UI.addButton("remove", this::remove);
62 UI.addButton("remove all", this::removeAll);
63 UI.addButton("reverse", this::reverse);
64 UI.addButton("shuffle", this::shuffle); */
65 UI.addButton("Testing", this::setTestList);
66 UI.addButton("Quit", UI::quit);
```

Code line 53
produced the first
Run show button

Code line 54
produced the
second button **Edit
show**

Code line 65
produced the third
button **Testing**

Code line 66
produced the 4th
button **Quit**

Class Edit Tools Options

SlideShow X

Compile Undo Cut Copy Paste Find... Close

Source Code ▾

```
111 /**
112  * Display just the current slide if the show is running.
113  * If the show is not running, display the list of images
114  * (as thumbnails) highlighting the current image
115  */
116 public void display(){
117     /*# YOUR CODE HERE */
118     if (showRunning){
119         UI.drawImage(images.get(currentImage), GAP, GAP, LARGE_SIZE, LARGE_SIZE);
120     }
121     else {
122         int left = GAP;
123         int top = GAP;
124         UI.clearGraphics();
125         for(int i = 0; i<images.size(); i++){
126             String fileName = images.get(i);
127             if (i == currentImage) {
```

```

public void drawString(String s, double left, double baseline)
public void drawImage(String filename, double left, double top)
public void drawImage(String filename, double left, double top, double width, double height)
    // erasing and inverting : all the drawXXX commands except drawImage also have an eraseXXX and invertXXX form
public void sleep(double millis) // pause program for specified time ( milliseconds ).
public void initialise ()      // ensure UI window has been initialised
public void quit ()          // delete UI window; usually halts the program.
    // Event-based input
public void addButton(String name, mth) // Add a button to input panel, mth is a method with no arguments
public void addTextField(String s, mth) // Add a textField to input panel, mth is a method with String argument
public void addSlider(String s, double min, double max, mth)
    // Add a slider to input panel, mth is a method with double argument
public void addSlider(String s, double min, double max, double initial, UISliderListener obj)
public void setMouseListener(mth) // Set MouseListener, mth is method with String and two doubles
public void setMouseMotionListener(mth) // Set MouseMotionListener,
public void setKeyListener(mth) // Set KeyListener for Graphics pane, mth is a method with String argument
// Typical usage:
    Ul.addButton("Load", this::loadData);
    Ul.addTextField("Name", this::setName);
    Ul.addSlider("Size", 1, 30, 5, this::setSize);
    Ul.addMouseListener(this::doMouse);
    Ul.addButton("Quit", Ul::quit);
public void loadData(){...
public void setName(String v){...
public void setSize(double v){...
public void doMouse(String action, double x, double y){
    if (action.equals("pressed")) {...
    else if (action.equals("released")){.....

```

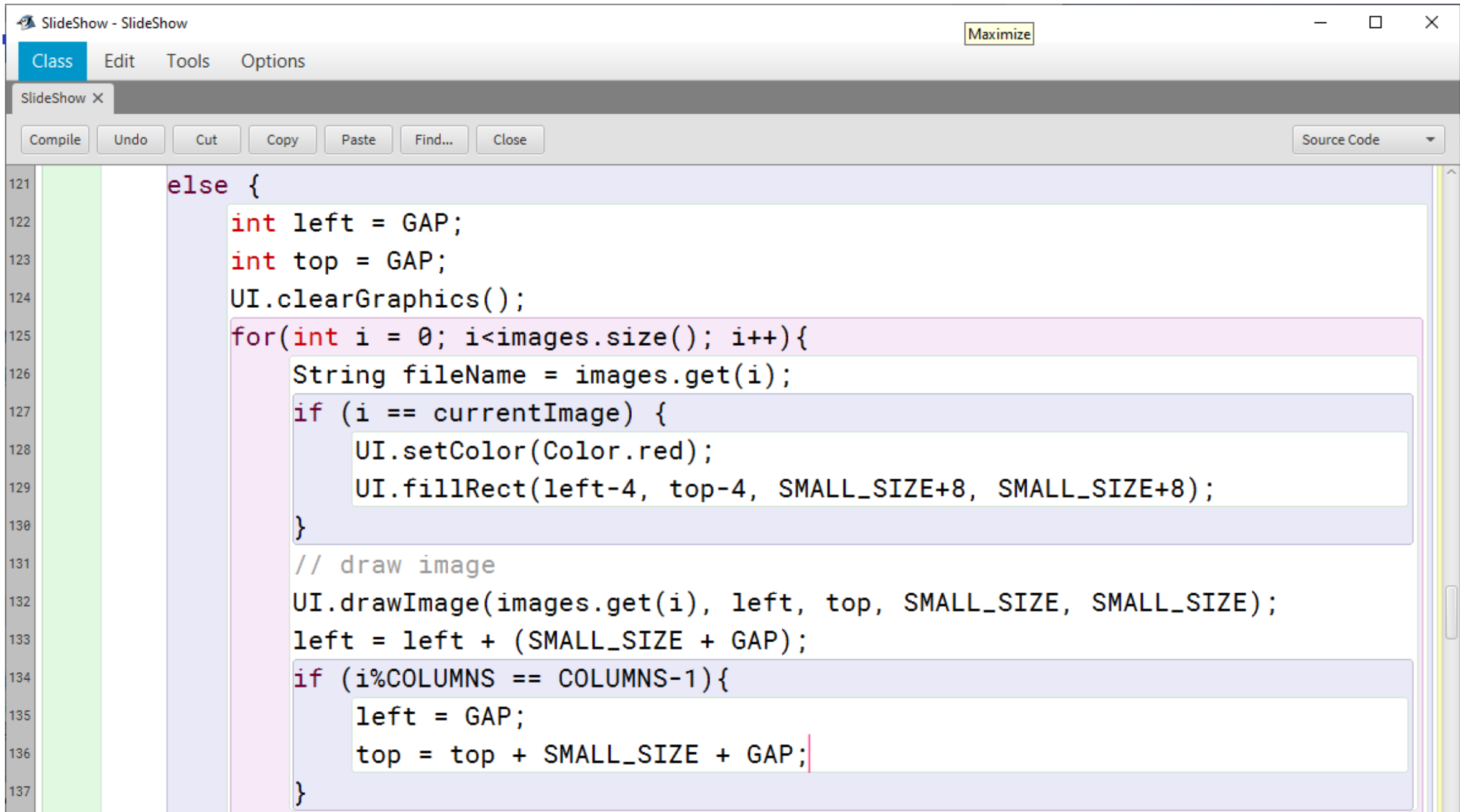
Class Edit Tools Options

SlideShow X

Compile Undo Cut Copy Paste Find... Close

Source Code ▾

```
111 /**
112  * Display just the current slide if the show is running.
113  * If the show is not running, display the list of images
114  * (as thumbnails) highlighting the current image
115  */
116 public void display(){
117     /*# YOUR CODE HERE */
118     if (showRunning){
119         UI.drawImage(images.get(currentImage), GAP, GAP, LARGE_SIZE, LARGE_SIZE);
120     }
121     else {
122         int left = GAP;
123         int top = GAP;
124         UI.clearGraphics();
125         for(int i = 0; i<images.size(); i++){
126             String fileName = images.get(i);
127             if (i == currentImage) {
```

```
121 else {
122     int left = GAP;
123     int top = GAP;
124     UI.clearGraphics();
125     for(int i = 0; i<images.size(); i++){
126         String fileName = images.get(i);
127         if (i == currentImage) {
128             UI.setColor(Color.red);
129             UI.fillRect(left-4, top-4, SMALL_SIZE+8, SMALL_SIZE+8);
130         }
131         // draw image
132         UI.drawImage(images.get(i), left, top, SMALL_SIZE, SMALL_SIZE);
133         left = left + (SMALL_SIZE + GAP);
134         if (i%COLUMNS == COLUMNS-1){
135             left = GAP;
136             top = top + SMALL_SIZE + GAP;
137         }
138     }
139 }
```



