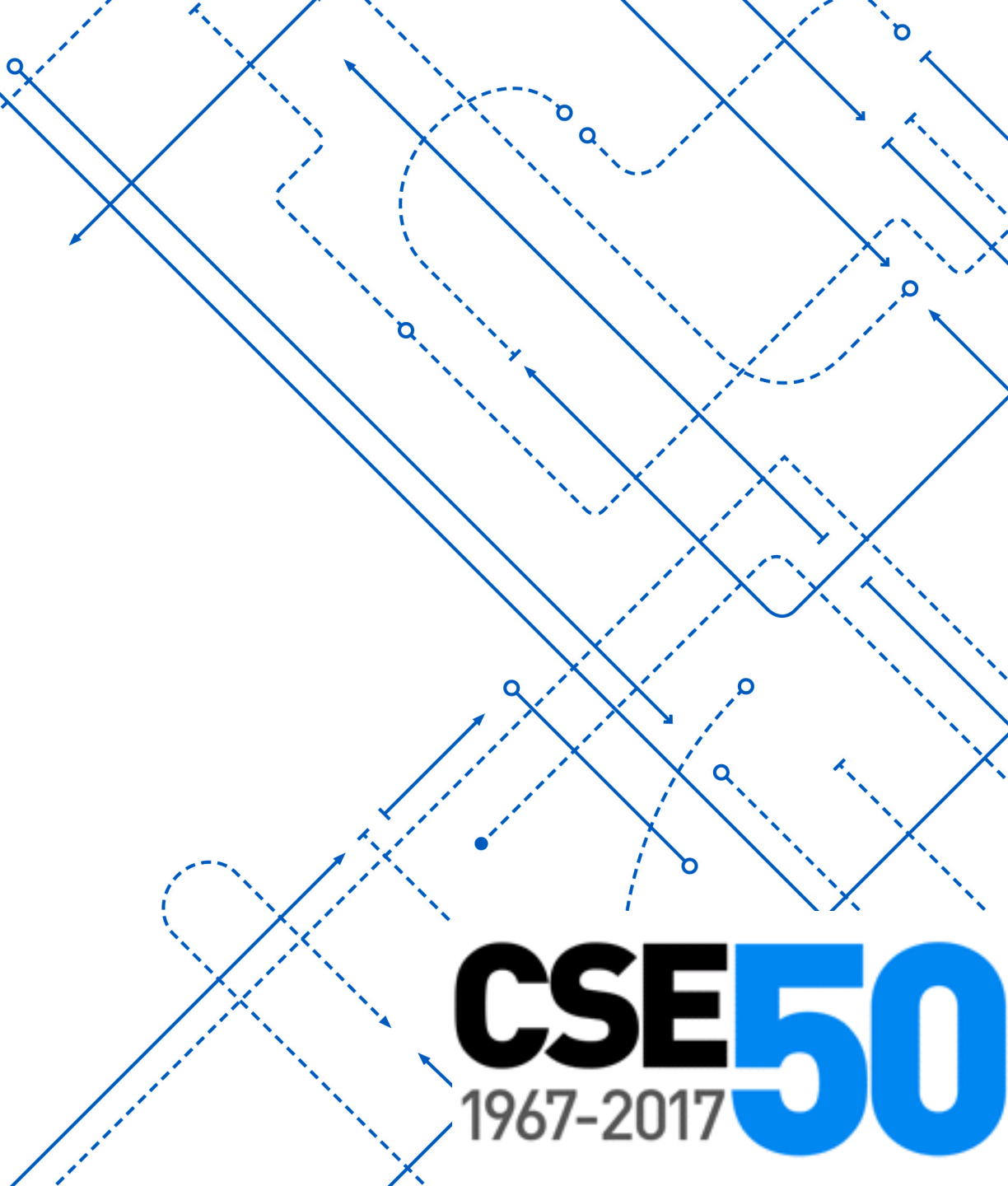
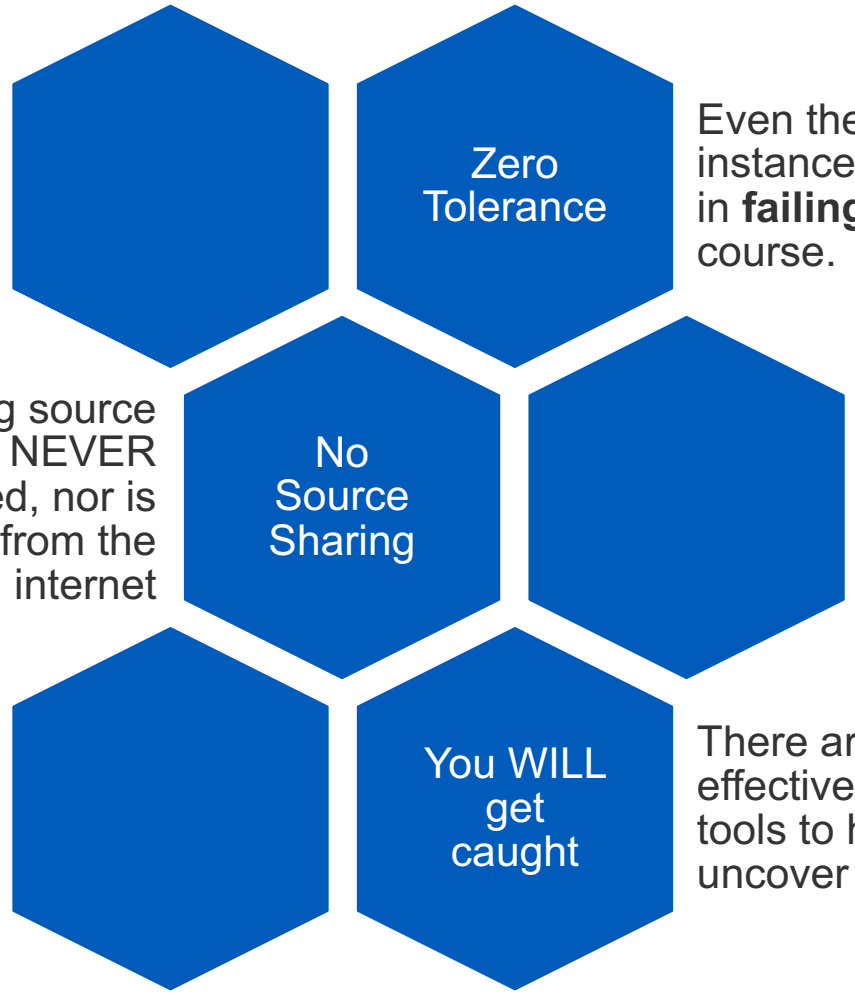
CSE 115 / 503
INTRODUCTION TO COMPUTER SCIENCE I

Dr. Carl Alphonce
Dr. Jesse Hartloff

University at Buffalo
School of Engineering and Applied Sciences



CSE 50
1967-2017

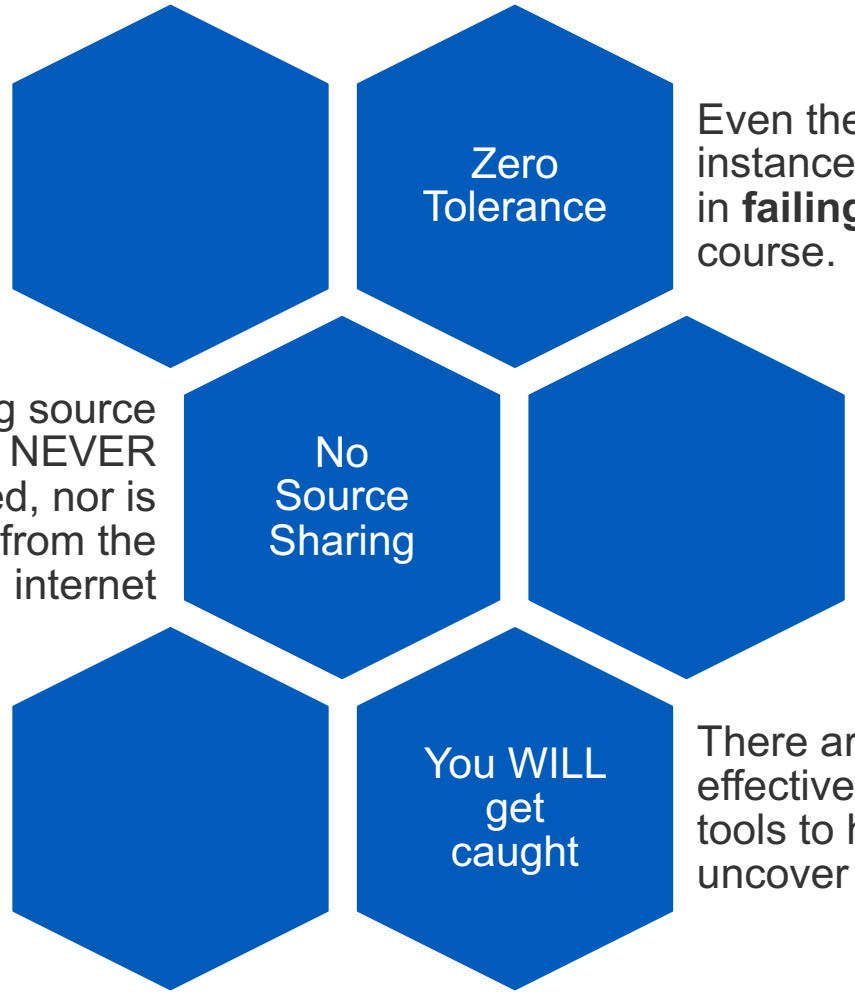


Even the first instance can result in **failing** the course.

Sharing source code is NEVER allowed, nor is getting it from the internet

There are very effective automated tools to help uncover violations!

Things that are OK	Things that are NOT
Group discussion (aside from exams)	Group collaboration on code
Properly cited references	Public Code
Discussing concepts with classmates	Sharing answers/solutions with classmates



Even the first instance can result in **failing** the course.

Sharing source code is NEVER allowed, nor is getting it from the internet

There are very effective automated tools to help uncover violations!

We are starting to run checks on existing and new submissions this week.

Things that are	Things that are NOT
ssion exams)	Group collaboration on code
Properly cited references	Public Code
Discussing concepts with classmates	Sharing answers/solutions with classmates

Physical attendance is required to earn points in:

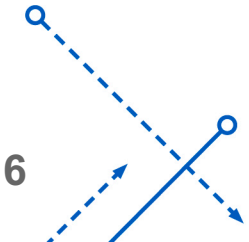
- Lecture (TopHat questions, Friday activity points)
- Recitation (Quiz points, Coding exercise points)
  - You must work on the lab machines, not your own (e.g. your laptop)

Graphical User Interface (GUI)

Using the Java graphics classes

In these slides we will explain the basics of how to create graphical programs.

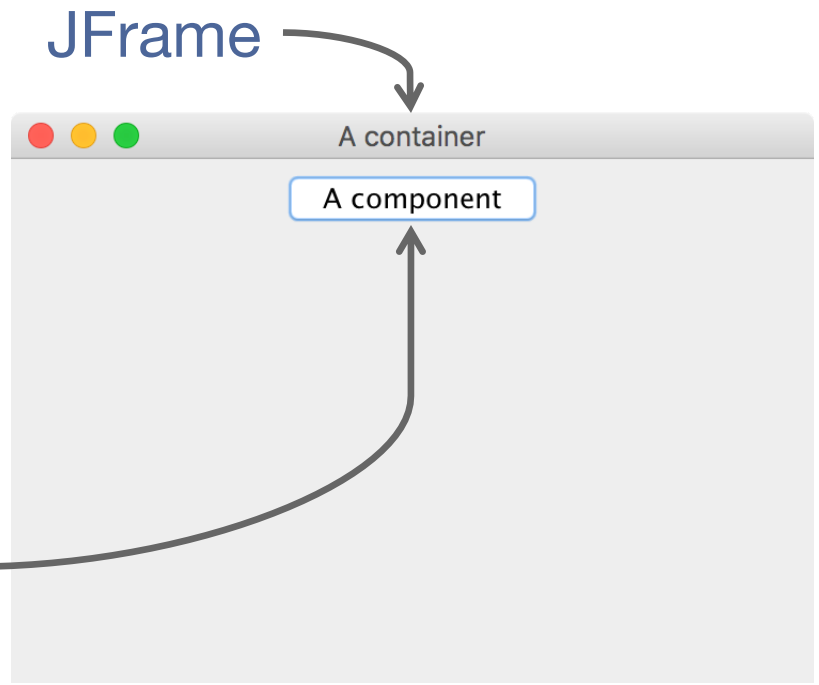
Some advanced issues will be glossed over (e.g. thread safety, graphics library design). They will be covered later in the course, or in later courses.



Graphical elements

There are two basic types of graphical elements:

- Containers
  - able to hold graphical objects, such as containers and components
- Components
  - must be put into containers
  - able to generate events when manipulated



Containers

Top-level containers

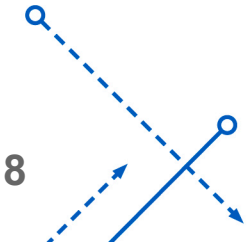
some containers are called “top-level” because they do not need to be placed inside any other containers

JFrame is a top-level container, meaning it can exist independently; a JFrame draws a window, complete with a title bar, scroll-bar, resize controls, etc.

Other containers (not top-level)

most containers must be placed inside some other container

javax.swing.JPanel is an example



javax.swing.JFrame

Top-level containers have multiple panes

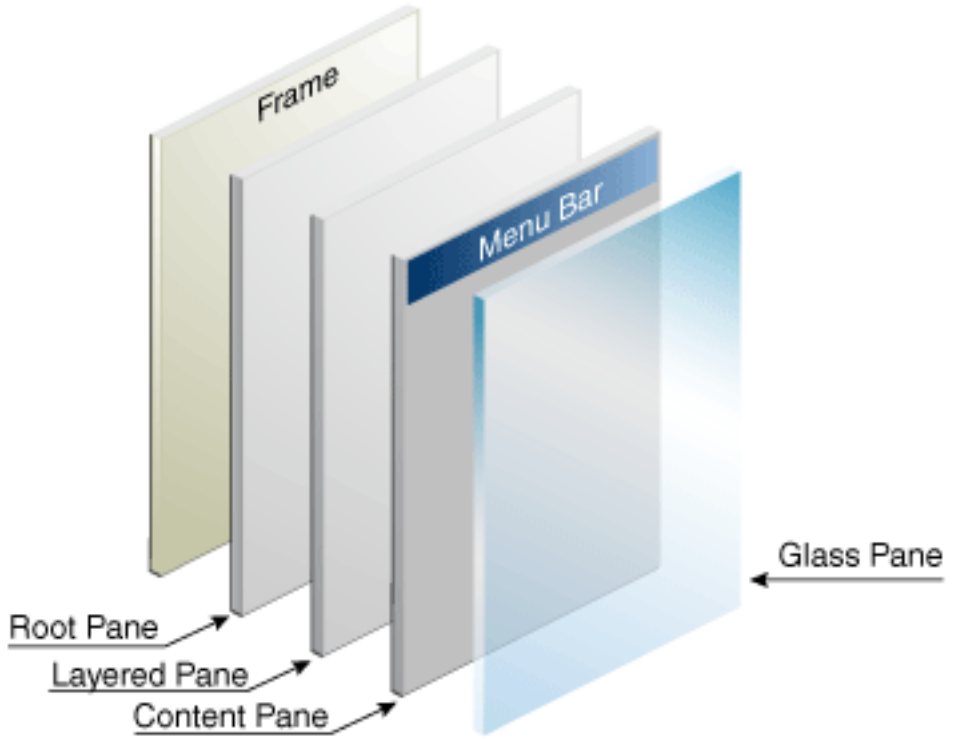


image credit:
http://docs.oracle.com/javase/tutorial/uiswing/components/toplevel.html

Example

Creating just a frame

new javax.swing.JFrame()

Creating a frame with a title

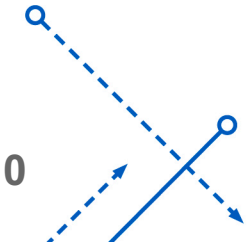
new javax.swing.JFrame(“My title”)

Making the frame visible

call setVisible(true) on the frame

Making application close when window is closed:

call setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) on the frame

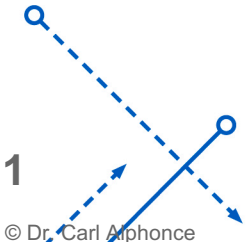


TopHat question

- Consider the following code:

```
JFrame window = new JFrame("Cool");
```

- What is the title that will appear on the JFrame?



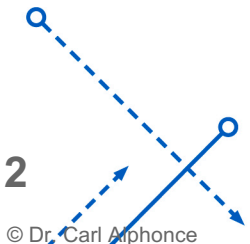
Components

- JLabel

  Can display text or an image or both.
  A non-reactive component.

- JButton

  Can display text or an image or both.
  A reactive component.



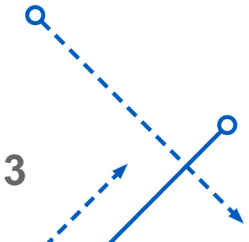
Adding components to a JFrame

We will add components to the content pane.

With javax.swing.JFrame, two ways:

- call `getContentPane()` on frame to get frame's content pane, then call `add(…)` on content pane to add a component
- call `add(…)` directly on the JFrame object

Second approach is just a convenience method, does the same thing the first approach.



Examples

```
JFrame window = new JFrame("Cool");
JLabel label = new JLabel("Info");
window.add(label);
```

```
JFrame window = new JFrame("Cool");
JLabel label = new JLabel("Info");
window.getContentPane().add(label);
```

TopHat question

- Consider the following code:

```
JFrame window = new JFrame("Cool");
JLabel label = new JLabel("Info");
```

- Which of the following are valid ways to add the JLabel label to the JFrame window?

```
contentPane.getJFrame().add(label);  // ANSWER 90

window.add(label); // ANSWER 17

label.add(window); // ANSWER 23

window.getContentPane().add(label); // ANSWER 68

Both answers 17 and 68 // ANSWER 45

Both answers 17 and 23 // ANSWER 90
```

Friday activity

- Monday
  - memory and object diagrams
  - variables, references, and objects
- Wednesday
  - Containers and Components
  - JFrame, JLabel