CS 342 Software Design

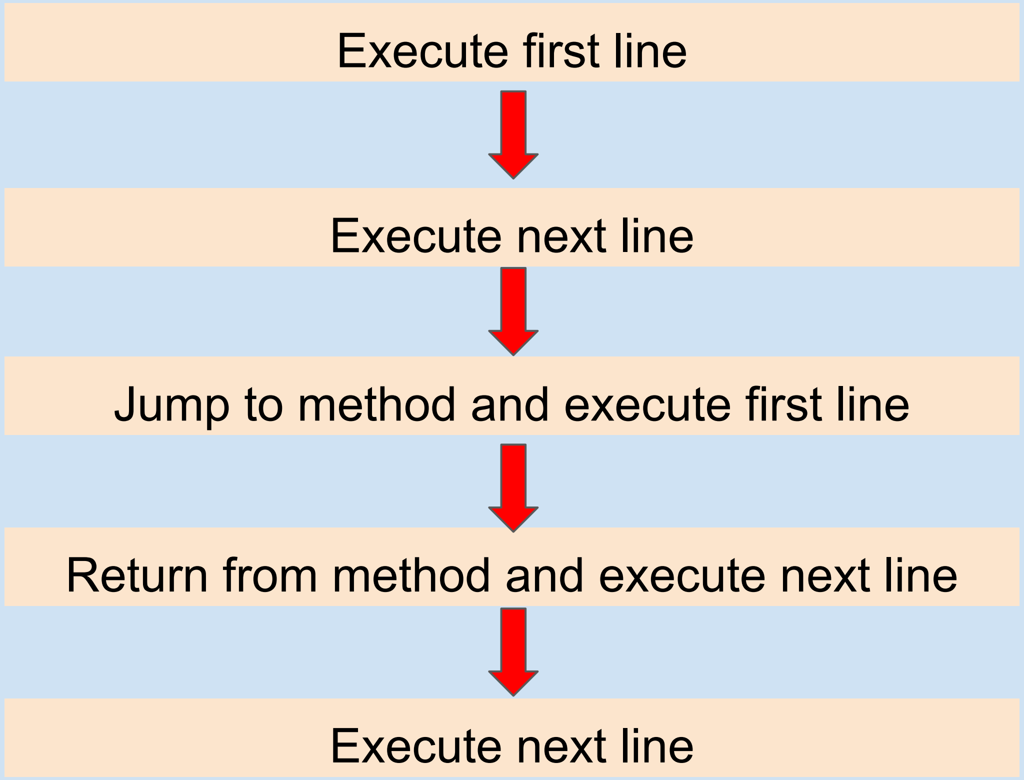
- Homework #2 is out and due Thursday
- Project #2 coming at the end of the week
- Java reference type: different than C++
- Event driven programing
- JavaFx

Clicker Question: What gets printed?

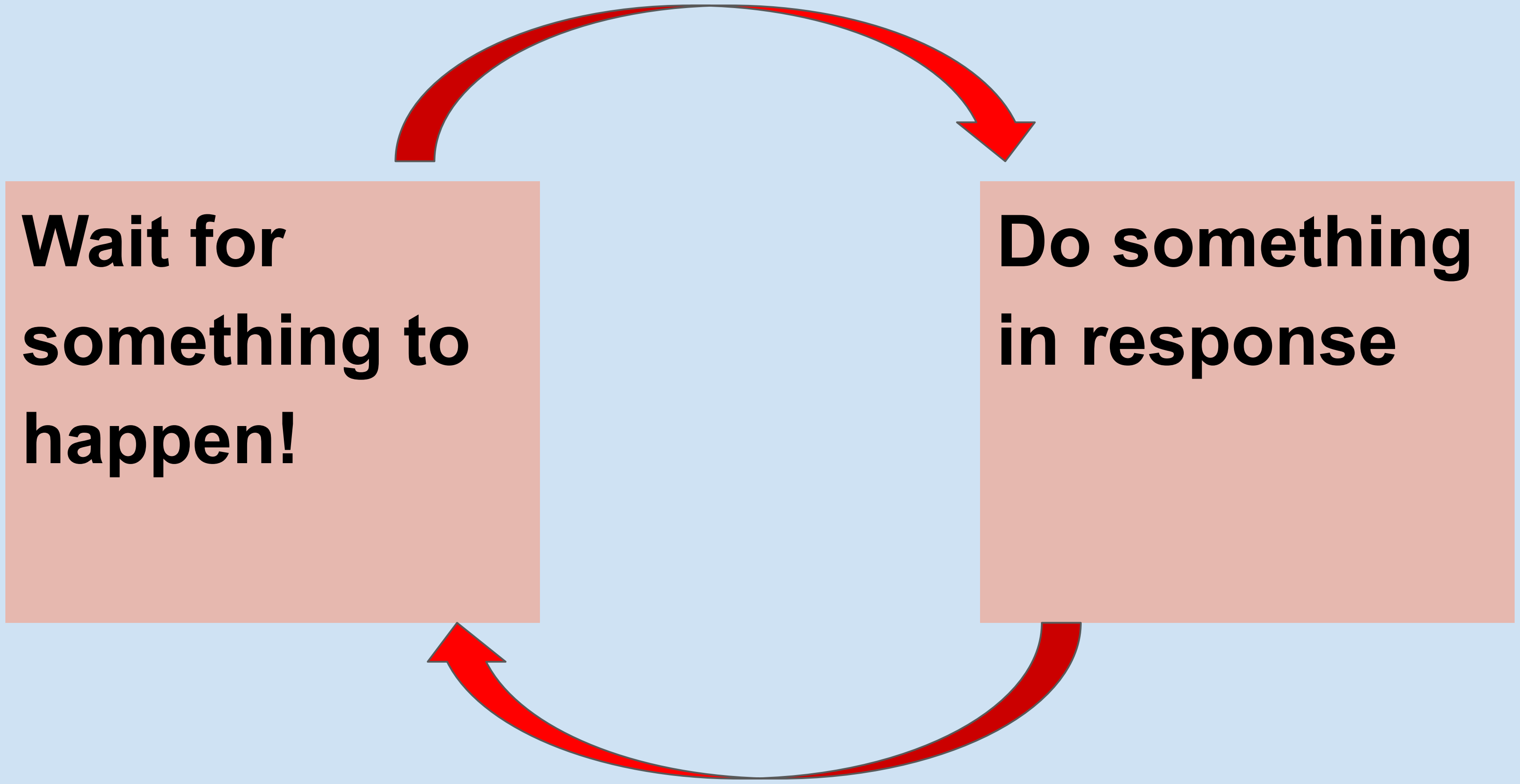
```
GenericQueue<String> q2 = new
GenericQueue<String>("Mark");

    Iterator<String> i2 = q2.createIterator();

    System.out.println( q2.dequeue() );

    i2.print();

    while(i2.hasNext()) {
        System.out.println(i2.next());
    }
```

A) Mark Null Mark
B) Null Mark Null
C) Mark Mark Null
D) Mark Null Null

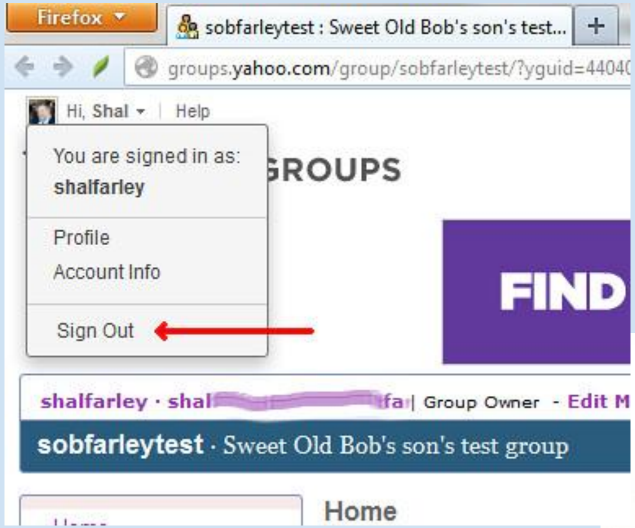
Procedural Programming: In Java, C, C++


Execute first line
Execute next line
Jump to method and execute first line
Return from method and execute next line
Execute next line

Event Driven Programming:



What are we waiting for?


Firefox
sobfarleytest : Sweet Old Bob's son's test...
groups.yahoo.com/group/sobfarleytest/?yguid=44040
Hi, Shal
Help
GROUPS
FIND
Profile
Account Info
Sign Out
Home

You are signed in as: shalfarley

shalfarley · shal[redacted]fa | Group Owner - Edit M

sobfarleytest · Sweet Old Bob's son's test group

The Event Loop:

Application starts with a single thread inside of one process.

The event loop starts (like an infinite loop) and is “listening” for events

If there is an event (button click, voice command, loading of a program….),

1. **Trigger the event handler (method) attached to that event**
2. **Handle the event**
3. **Return to “listening” for events**

****The event loop stops when the process stops****

What is JavaFx?

- Rich Internet Application: Adobe Flash, Microsoft SilverLight, JavaFX
- Written in Java and platform independent: browser penRate 76%
- JavaFX can run on various devices such as Desktop Computers, Mobile Phones, TVs, Tablets, etc.
- Some IDE’s provide a drag and drop interface
- Rich set of API’s to develop GUI applications, 2D and 3D graphics, etc.
- Access the features of Java languages such as Generics, Annotations, Multithreading, and Lambda Expressions
- Optimizes for graphics card but defaults to software rendering stack

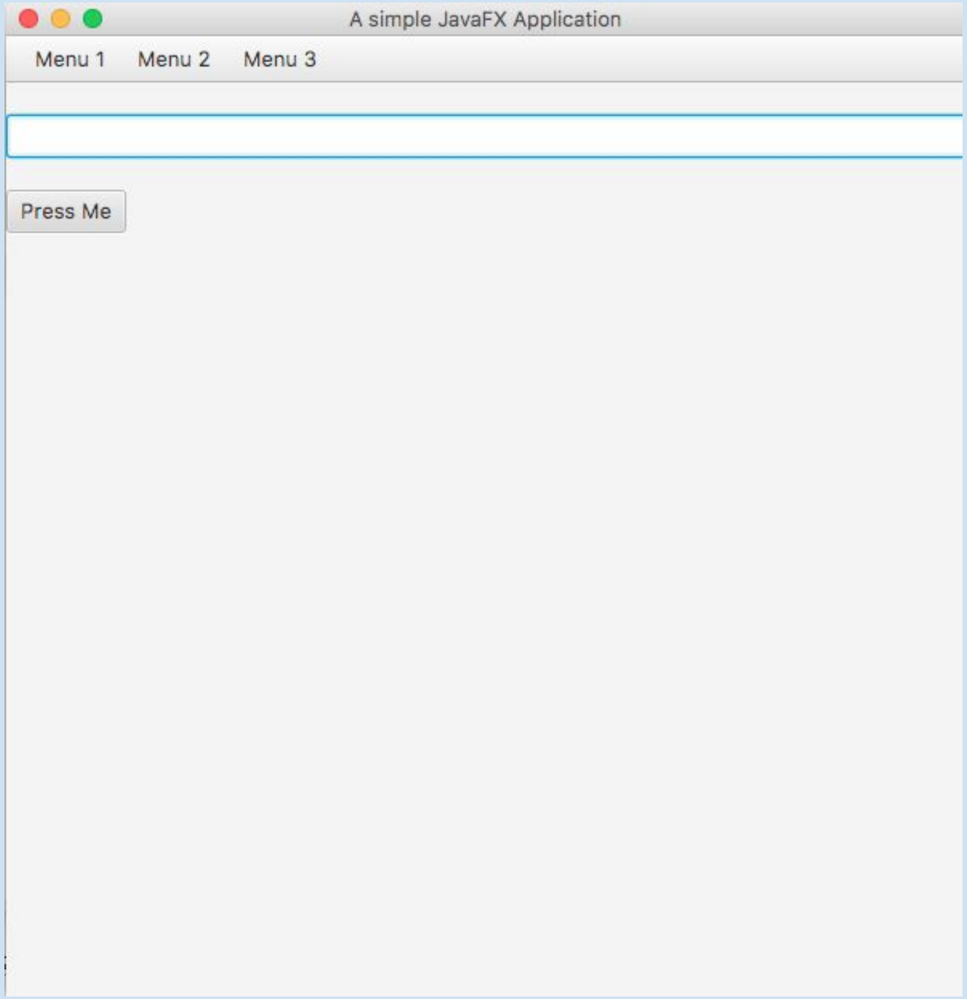
JavaFX application lifecycle:

1. Constructs instance of the specified Application class
2. Calls the init() method
3. Calls the start() method
4. Waits for the application to finish
    a. Application calls Platform.exit()
    b. Last window is closed
5. Calls the stop() method

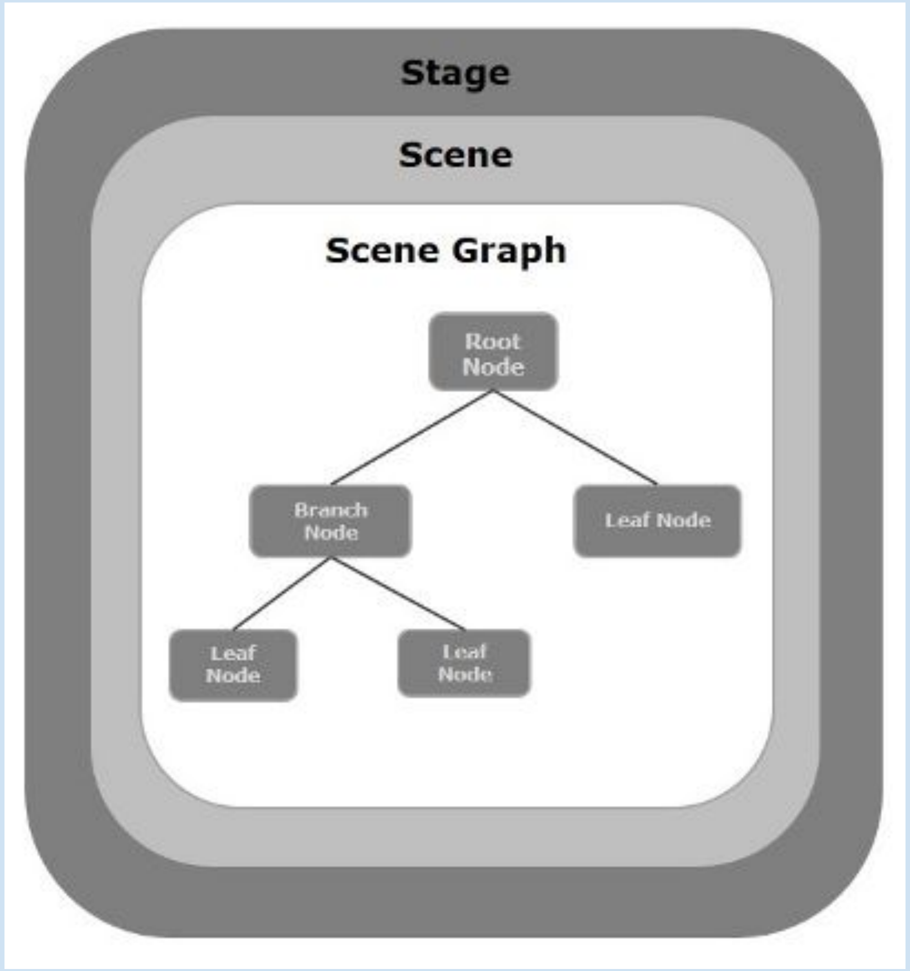
Application runs on a single thread...Application Thread

init() runs on Launcher Thread

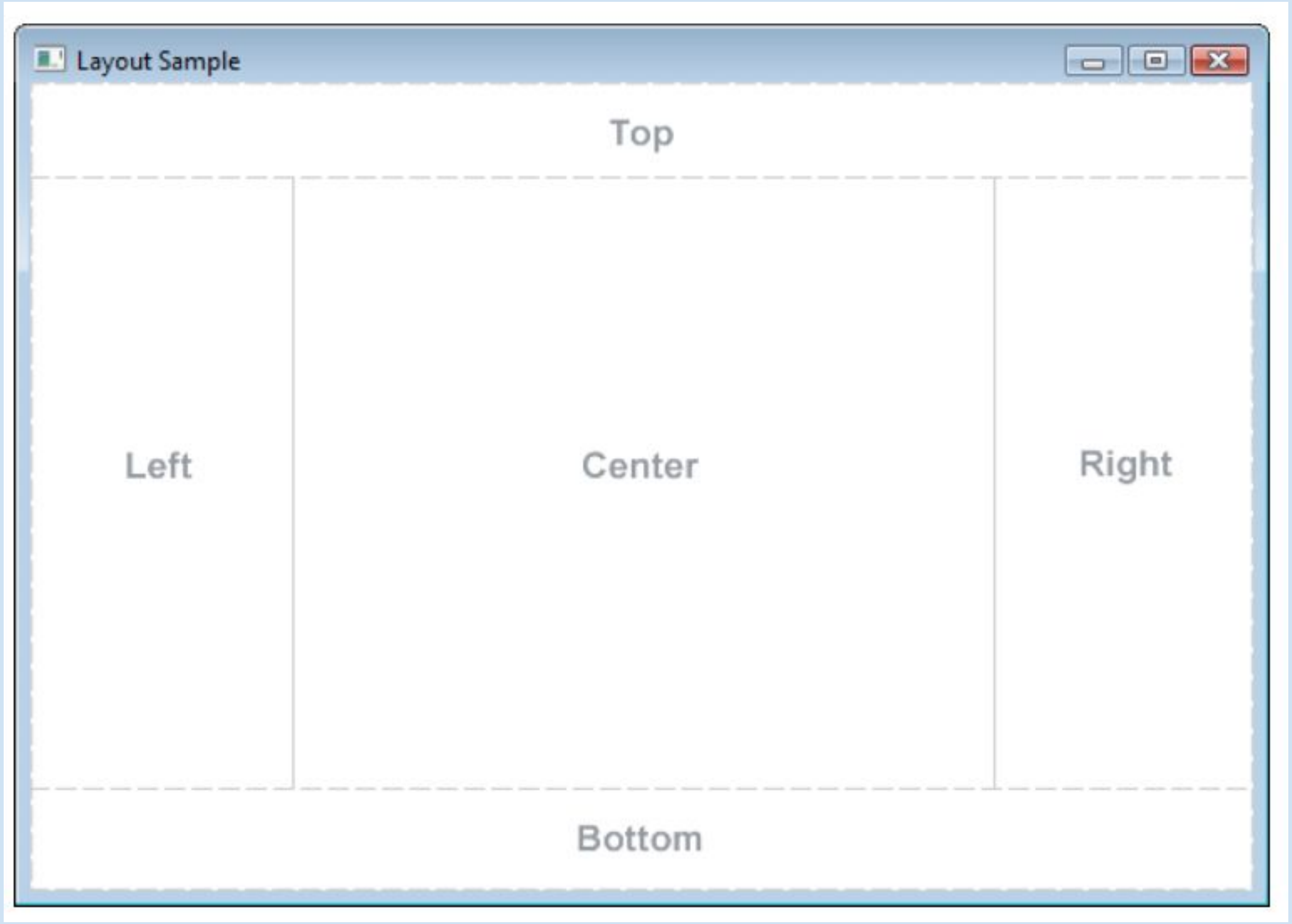
1st Application:



Application Structure:



BorderPane:



Anonymous Classes:

Declare and instantiate a class at the same time.

Makes code more concise.

Used all the time for handlers in GUI programing.

It is an expression: new operator, name of class, constructor for that class, body with method declarations.

Functional Interface:

A functional interface is any interface that contains only one abstract method. (A functional interface may contain one or more default methods or static methods.) Because a functional interface contains only one abstract method, you can omit the name of that method when you implement it. To do this, instead of using an anonymous class expression, you use a lambda expression.

Write your own or use one from java.util.function!

EventHandler<ActionEvent>

This is a functional interface!

https://docs.oracle.com/javase/8/javafx/api/javafx/event/EventHandler.html

So, we can just use a lambda expression! No need for extra syntax or anonymous class

Lambda Expressions:

- Enable to treat functionality as a method argument, or code as data.
- A function that can be created without belonging to any class.
- A lambda expression can be passed around as if it was an object and executed on demand.

Lambda Expressions:

Lambda expressions express instances of functional interfaces (An interface with single abstract).

lambda expressions implement the only abstract function and therefore implement functional interfaces.

Lambda Syntax:

- Comma separated list of parameters: (x,y,z)
  - You can omit data type.
  - Omit parentheses if only one parameter
- Arrow: ->
- Body: single expression or a statement block.

Let’s see an example!

JavaFX CSS

How to style your application:

https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html#typecolor