CS 180 Problem Solving and Object Oriented Programming

Fall 2011

http://www.cs.purdue.edu/homes/apm/courses/CS180Fall2011/

Notes for Week 9:

Oct 17-21, 2011

This Week:

10/17-20

1. Creating GUIs
2. Frames, Panels, Buttons
3. Text fields
4. Listeners
5. Layouts
6. Method signatures
7. Abstract class, interface

Aditya Mathur
Department of Computer Science
Purdue University
West Lafayette, IN, USA

Announcements

1. Homework 6 due Thursday-Friday.
2. Project 3 assigned. Form teams of up to 3.
3. You may work independently though not recommended.
4. Having trouble with finding a partner? Send mail to Dr Lorenzo Martino for help.

   Lmartino@purdue.edu
5. Programming competition round 1:

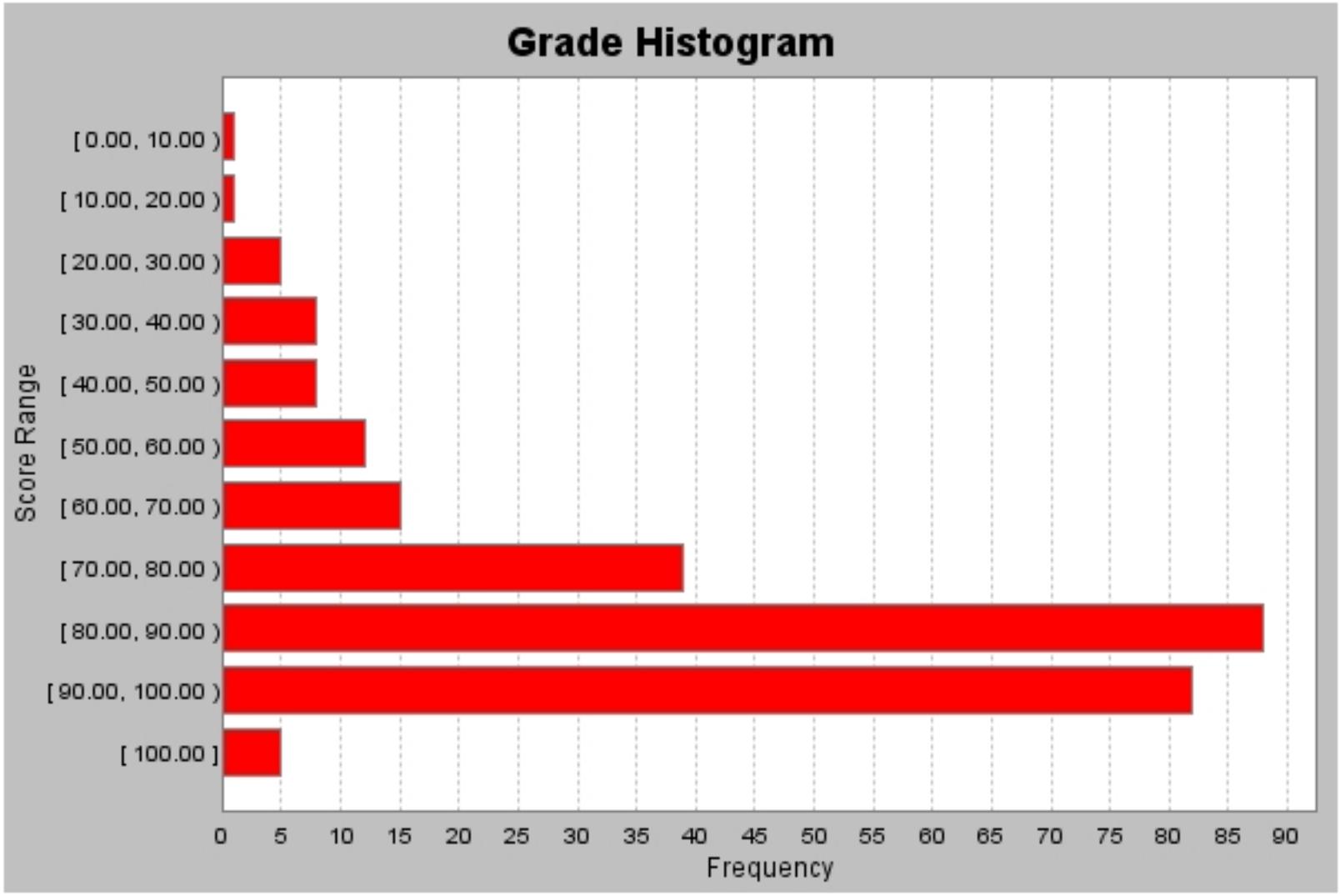
   Saturday October 22, 6pm, LWSN 3102AB.

Exam 1: Letter grades and statistics

	High/Low
A+	100/98
A	97/94
A-	93/90
B+	89/87
B	86/82
B-	82/78
C+	78/73
C	72/67
C-	66/51

Average:	79.5
Median:	86.0
Maximum:	100.0
Standard Deviation:	18.15

Exam 1: Letter grades and statistics



Readings and Exercises for Week 9

Readings:

GUI: 13.2, 13.3, 13.4

Exercises:

13.1, 13.6, 13.7

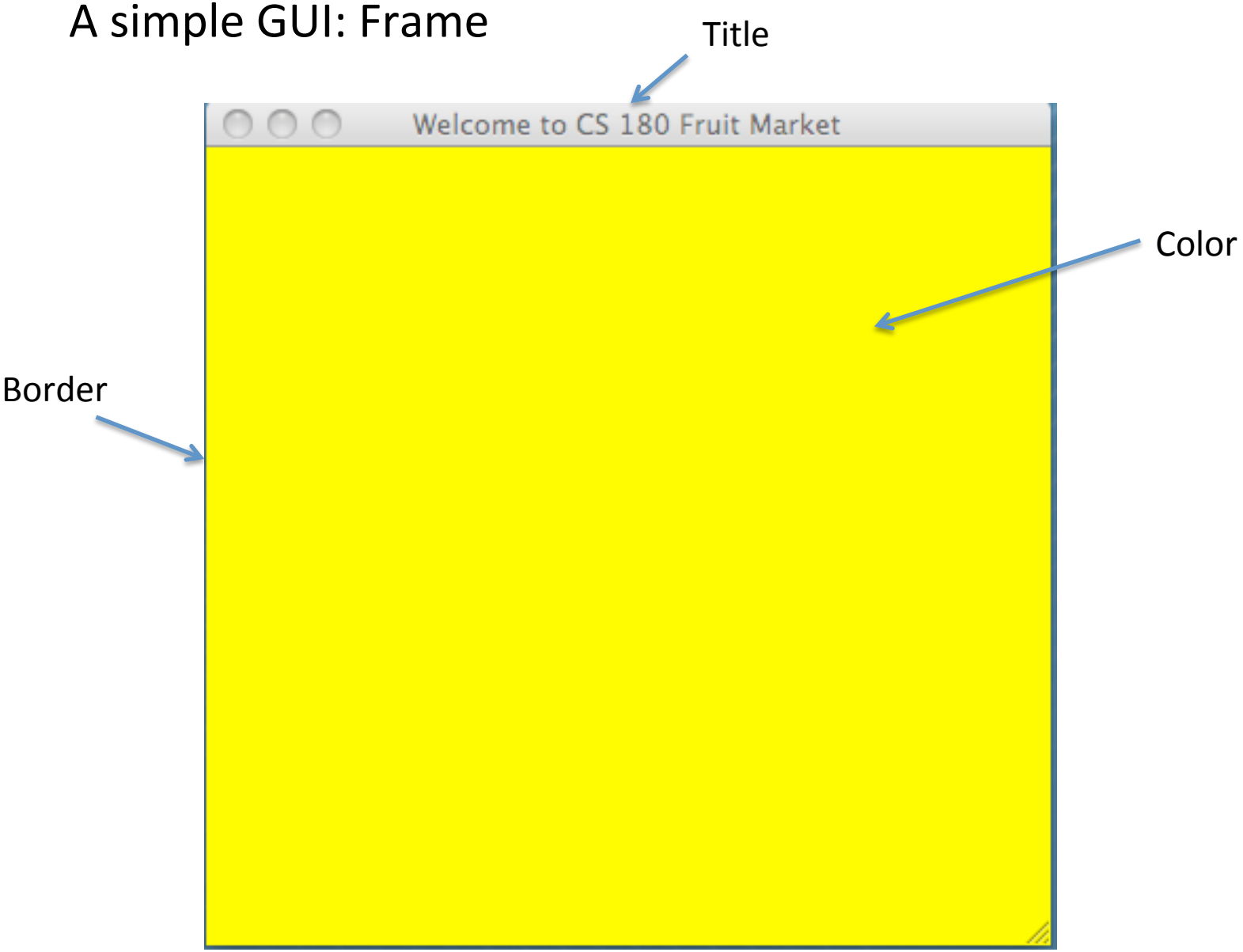

BlackBerry
12:21 PM
THU, MAY 12
at&t
AT&T
EDGE
Messages
SPACE
SYM
at&t

GUIs

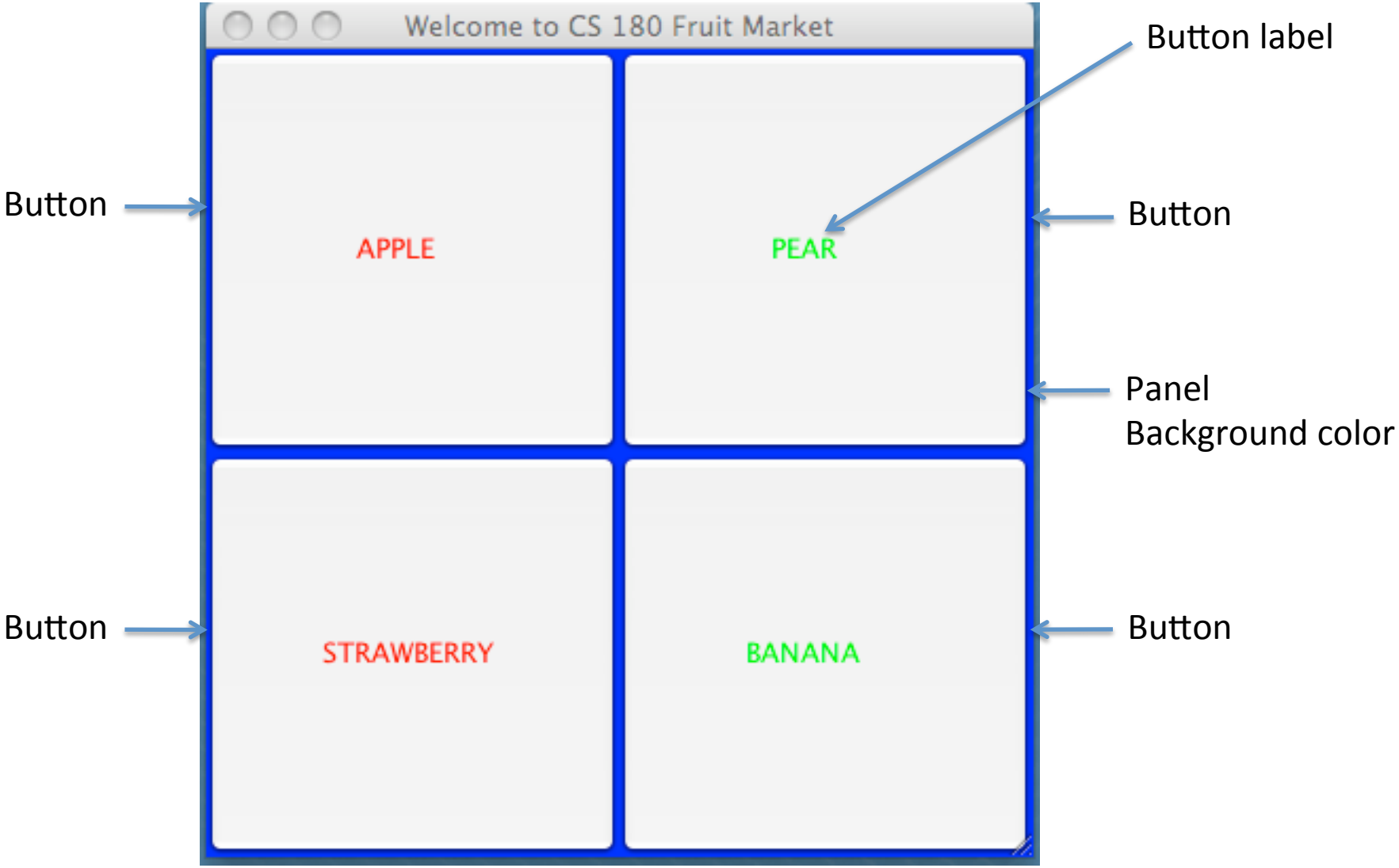

MOTOROLA
3:00 PM
See all your apps.
Touch the Launcher icon.
verizon


9:41 AM
Messages
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Notes
Utilities
iTunes
App Store
Settings
Phone
Mail
Safari
iPod

A simple GUI: Frame



A simple GUI: Frame with a Panel and Four Buttons



Problem 1

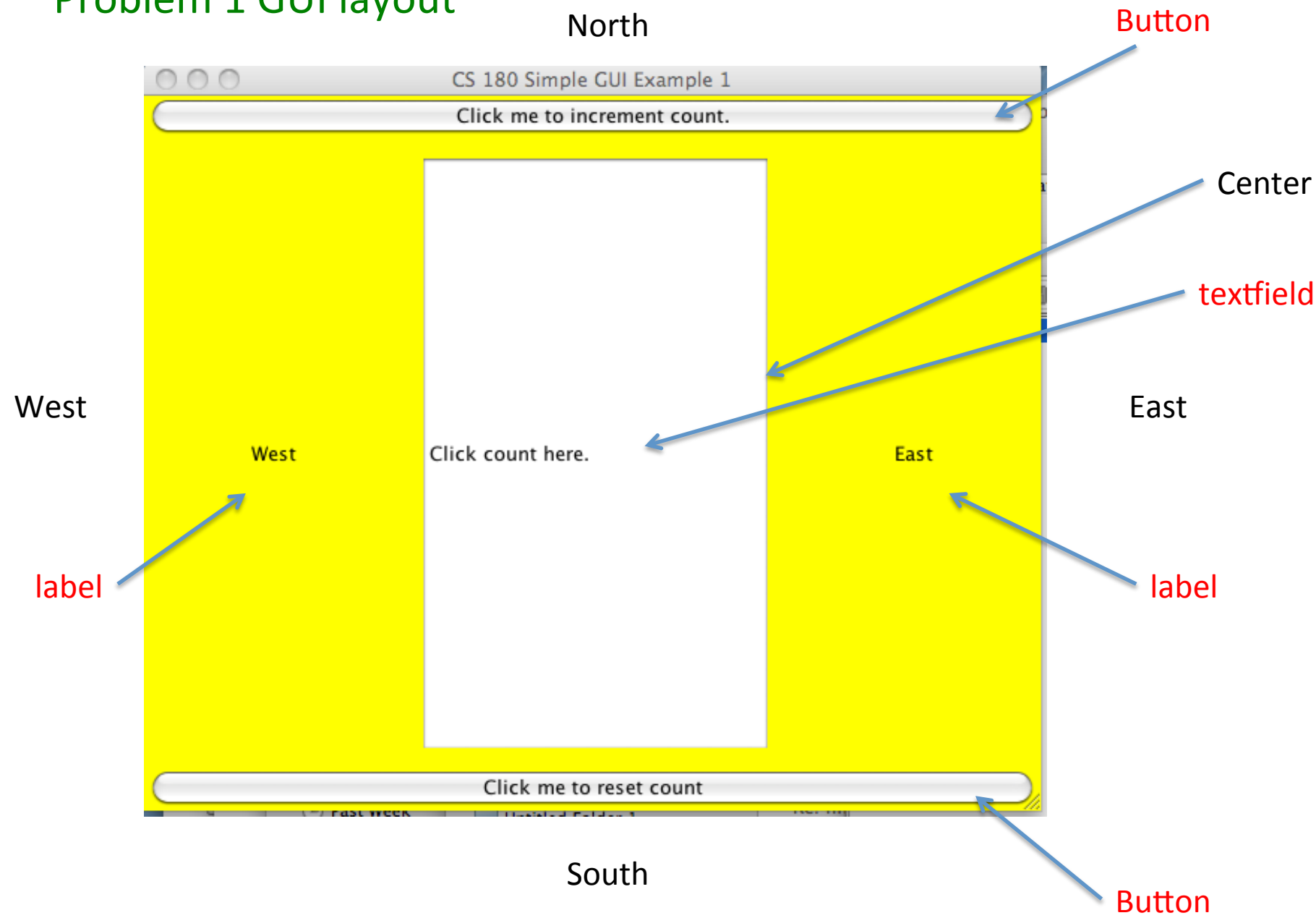
Write a program to generate a GUI shown on the next slide. The GUI has two buttons, two labels, and one text box.

When the button on the North is clicked the new count of clicks should be displayed in the center.

When the button located South is clicked, the count should be reset to 0.

When the mouse enters (or exits) the west or the east area, a suitable message should be displayed in that area.

Problem 1 GUI layout



Live demo: Example 1

Announcements

1. Feast with Faculty: Today: 6:30pm Ford Dining Hall
2. if (you are in SCI 210)

   work in a team for Project 3.

   }
3. Programming competition round 1:

   Saturday October 22, 6pm, LWSN 3102AB.

Quiz: 10/19/2011

Q1. float [] a=new float[100];

Number of elements in array a is

(a) 101

(b) 100

(c) 99

(d) 102

Q2. int [] a=new int[100];

Correct values of the index into a range from

(a) 0 to 100

(b) 1 to 100

(c) 0 to 99

(d) 1 to 99

Q3. boolean [][] a=new boolean [5][3];

Maximum number of values of type boolean that can be stored in array a is

(a) 5

(b) 8

(c) 3

(d) 15

Q4. int [][] a=new int [5][];

Number of columns in a is

(a) 5

(b) undefined

(c) 0

(d) 1

Q4.
```
float [] a=new float [5];
for (int r=0; r<6; r++){
    a[r]=7;
}
```

Which statement is correct?

(a) Elements in a are initialized to 7

(b) There will be an index out of bounds exception

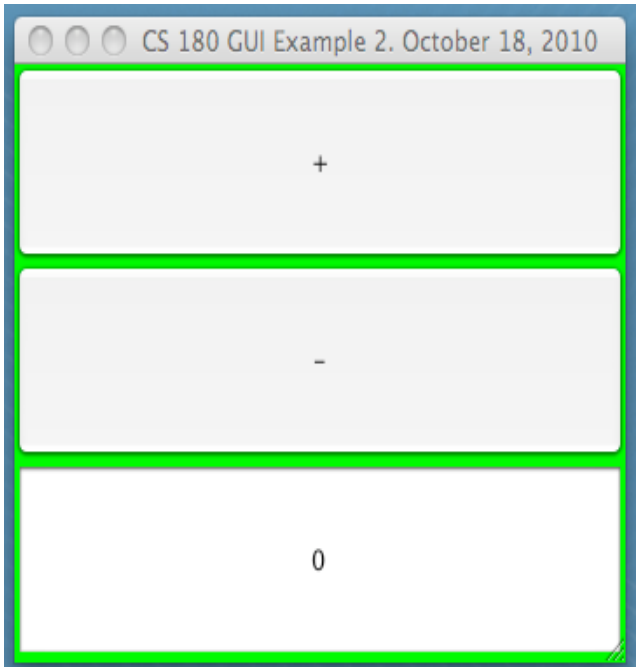
(c) 7 cannot be assigned to a variable of type float

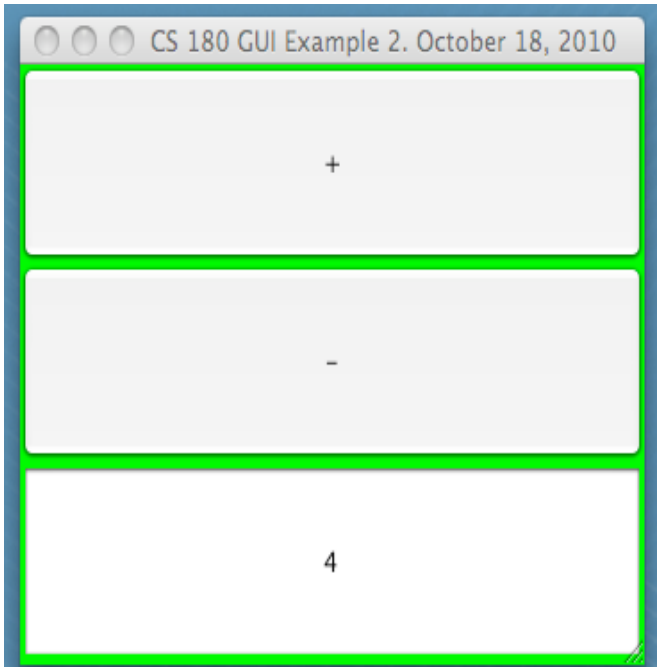
(d) r must be of type float

End of Quiz: 10/19/2011

Problem 2

Write a program to generate a GUI shown below. The GUI has two buttons labeled + and – and a text box. Clicking the plus button adds to the click count and clicking the minus button decrements it. The count is displayed in the text box.



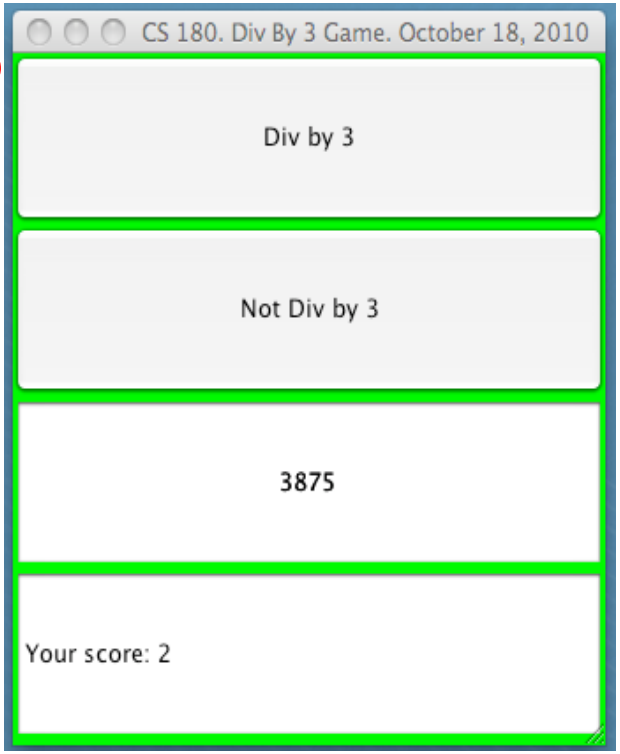


Live demo: Example 2

Problem 3

Write a math game program to generate a GUI shown below. The GUI has two buttons labeled Div by 3 and Not Div By 3 and two text boxes. A random integer is displayed in one textbox and the player must decide whether or not it is divisible by 3. Score is displayed in the other text bo

The game never ends unless the program is forcefully terminated.



Live demo: Example 3

Problem 3 based exercise

Modify the Divide by 3 game so that it displays the total duration of the game in minutes and seconds in a separate text box.

Methods, method, signatures, Interfaces and abstract classes

More GUI

Method declaration

Syntax:

```
modifier  type name (parameters){
        body consisting of 0 or more statements.
};
```

Examples:

```
public String getModel (); // return string, no parameters

public boolean search(int []a,  int n); // return boolean, two
        parameters

public void doIt(); // No return value, no parameter
```

Method call: Examples

Declaration: public String getModel (){ …. }

String s=getModel();// Return value assigned to s.

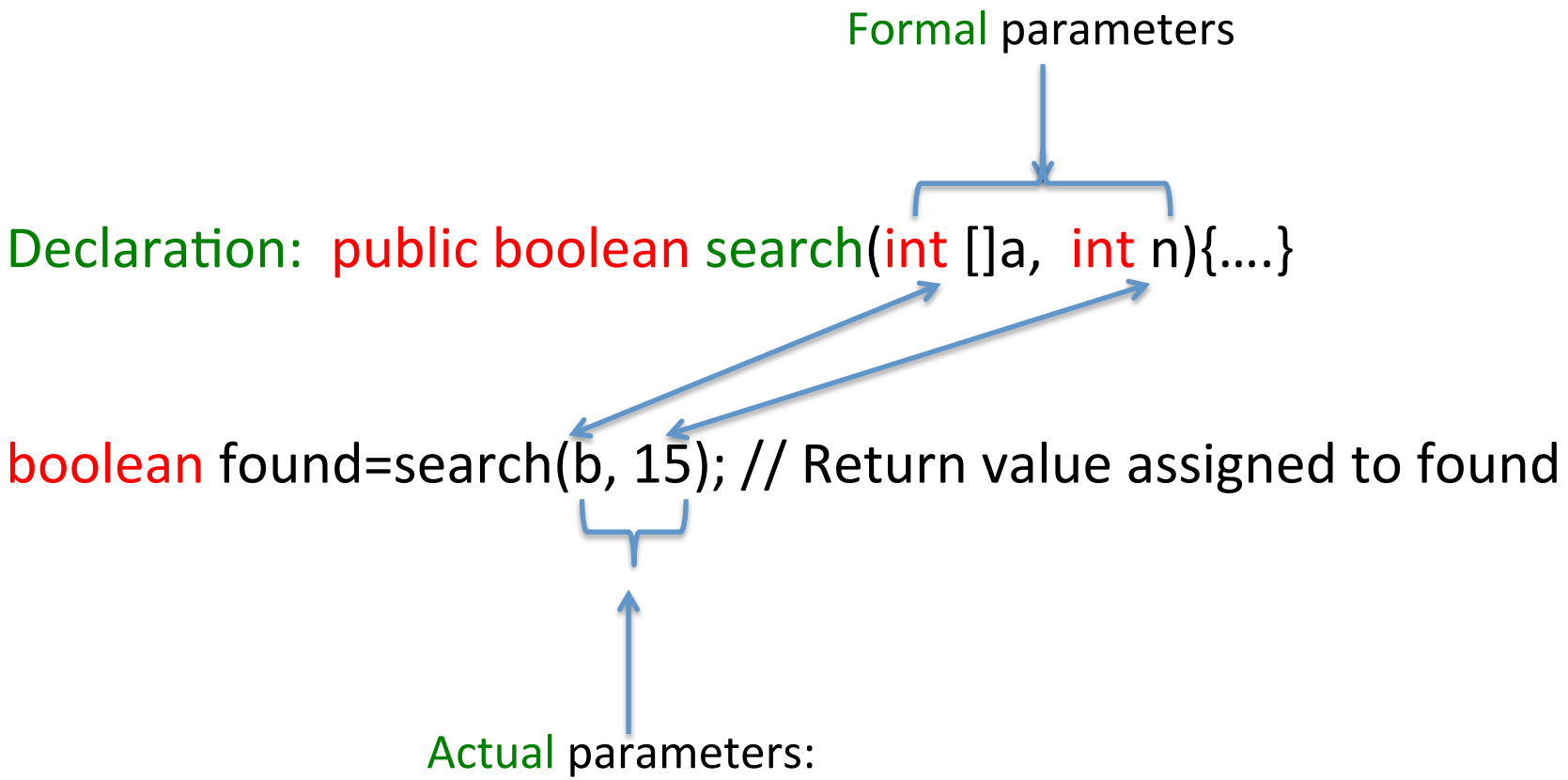
Declaration: public boolean search(int []a, int n){….}

boolean found=search(b, 15); // Return value assigned to found

Declaration: public void doIt();{ ….. }

doIt(); // No return value.

Method call: Parameter correspondence

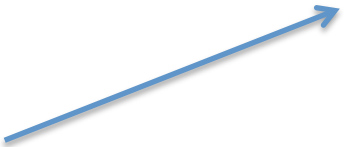


Method call: Parameter passing

Formal parameters

Declaration: public boolean search(int []a, int n){….}

boolean found=search(b, 15); // Return value assigned to found

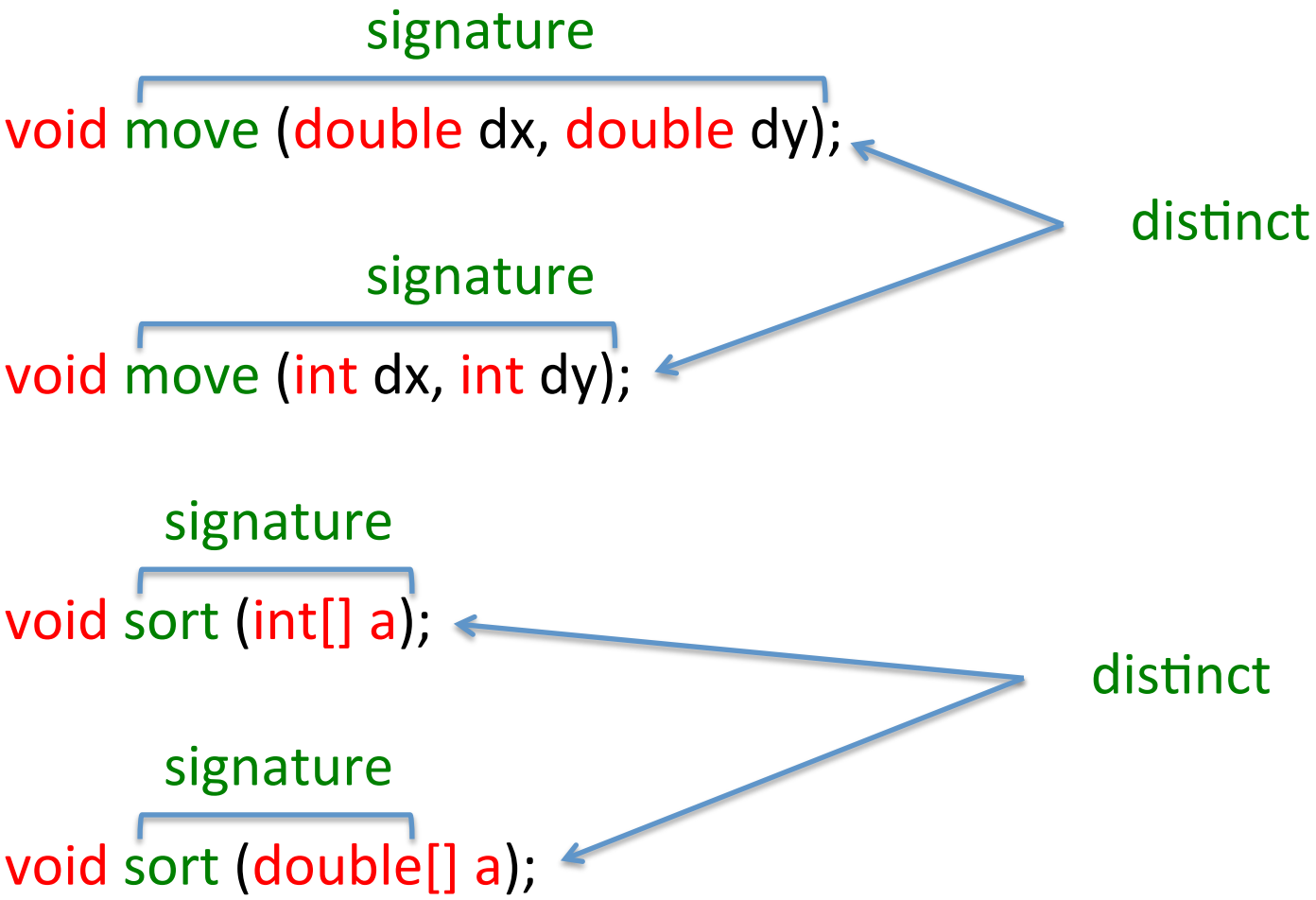


Actual parameters:

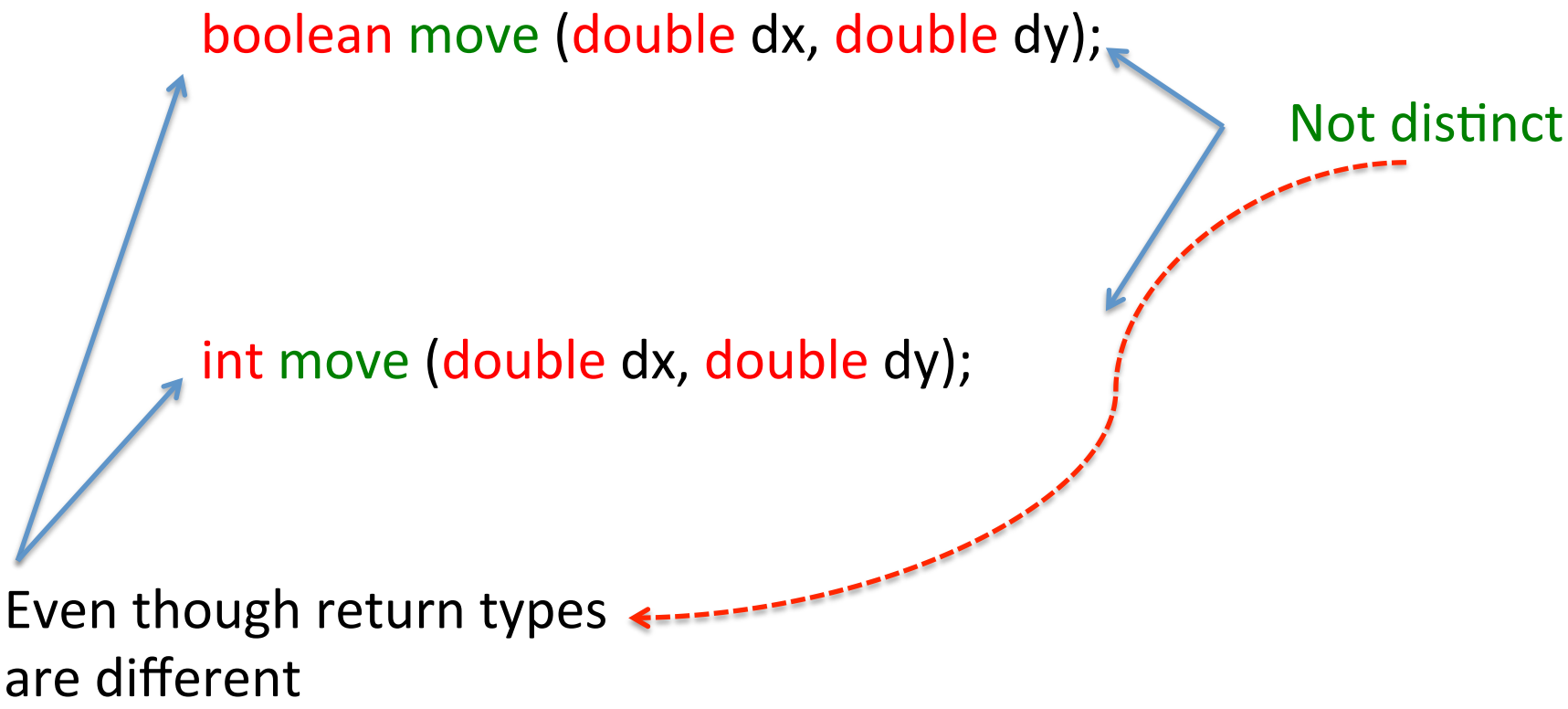
- Reference to array b is passed to search; thus, inside search a reference to a is actually a reference to b.

- 15 is passed to search for n. Thus, inside search n is 15.

Method signatures: name and parameters: Yes



Method signatures: return type: NO



Classes, Interface, and Abstract Classes

Class

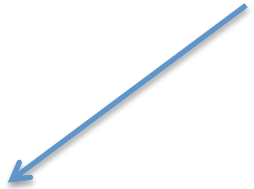
Constructor, methods; used to create objects

Interface:

- Class with only method signatures (abstract methods)
- An interface does not implement any method
- Methods are implemented by a class that uses the interface
- Thus, multiple implementations could exist

Interface: Example 1

Abstract methods; not implemented



```
interface Car{
   void cruise(double speed); // cruise at speed
   void startEngine(); // Start car engine
   void slowDown(double speed); // Slow down to speed
   double getSpeed(); // Returns current speed
   String getLicense(); // Returns license plate number
}
```

Interface: Example 2

```
interface ActionListener{
    void actionPerformed(ActionEvent);
}
```

A user class implements an ActionListener.

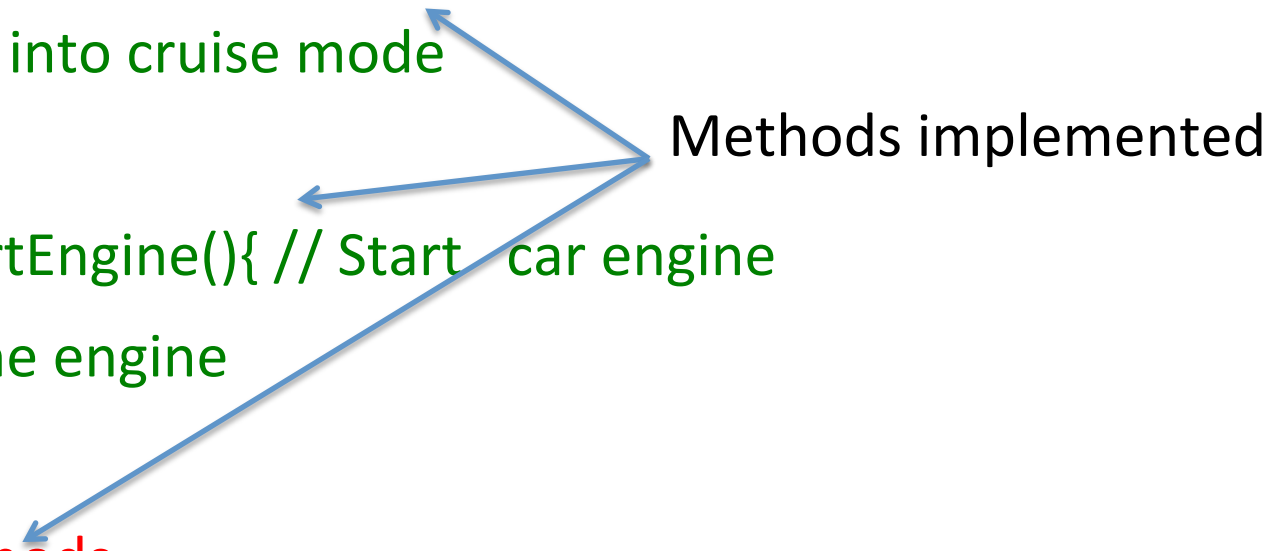
Interface: Example 3

```
interface MouseListener{
    void mouseClicked(MouseEvent e)
    void mouseEntered(MouseEvent e)
    void mouseExited(MouseEvent e)
    void mousePressed(MouseEvent e)
    void mouseReleased(MouseEvent e)
}
```

A user class iimplements a MouseListener.

Interface: Example 1 : Implementation

```
public class MyCar implements Car{
    public void cruise(double speed){ // cruise at speed
// Code to get car into cruise mode
}
    public void  startEngine(){ // Start   car engine
// Code to start the engine
}
……  // Other methods
}
```



Interface: When and why?

- Use an interface to specify a contract between two parties.
- When a team is developing an application, a core group of people can specify interfaces while other groups are free to implement these as they consider appropriate.
- Example: A car manufacturer can specify an interface that will be used by software developers of all models made by this manufacturer.
- Interfaces allow specification of uniform and contractual obligations across several products.

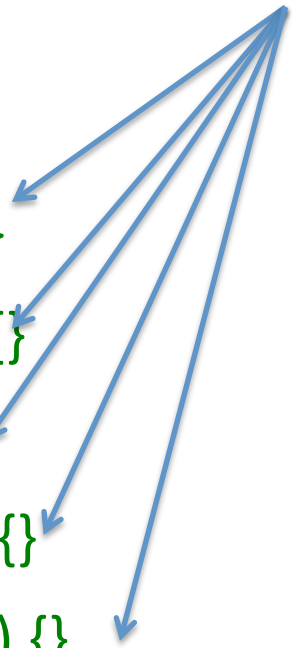
Abstract class

Similar to interface but may:

- Implement zero or more methods
- Provides abstract methods
- An abstract method is one with only a signature but no implementation
- A Java class can extend an abstract class by implementing one or more of its abstract methods.

Abstract class: Example

Empty implementations that may be overridden



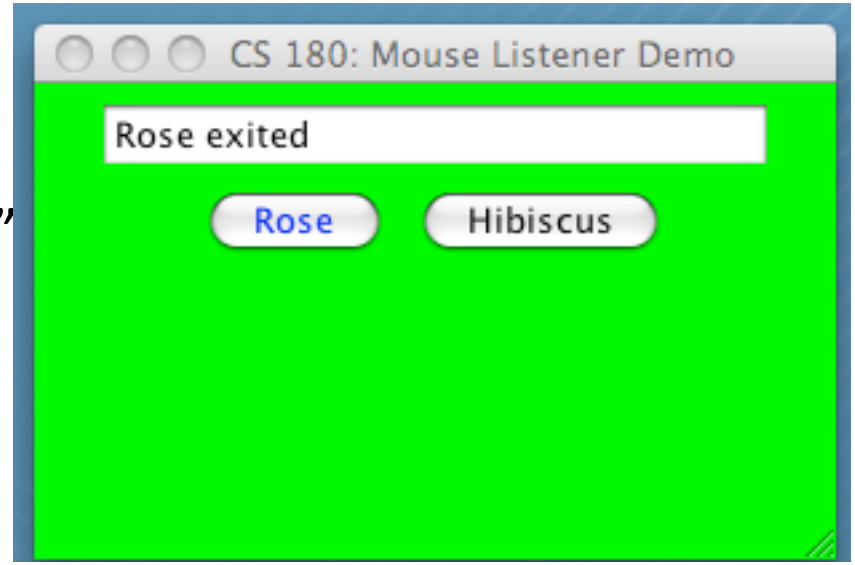
```
public abstract class MouseAdapter{
    void mouseClicked(MouseEvent e){}
    void mouseEntered(MouseEvent e){}
    void mouseExited(MouseEvent e){}
    void mousePressed(MouseEvent e) {}
    void mouseReleased(MouseEvent e) {}

}
```

A user class extends a MouseAdapter.

Problem

Write a program to generate a GUI that has two buttons labeled Rose and Hibiscus and a text box. When the mouse enters Rose, the text box should display "Rose entered" and similarly for Hibiscus. When the mouse exits a button a similar message should be displayed.

When a button is clicked, the Display should be "Rose clicked" and similarly for Hibiscus.



Week 9: October 17-21, 2011
Hope you enjoyed this week!

Questions?

Contact your recitation instructor. Make full use of our office hours.