Computer Programming Fundamentals

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/

ASSIGNMENT 4

- Classes and objects
- Due Friday 10/8

POST TOPICS FOR DEBUGGING + MIDTERM REVIEW

TALK TODAY AT 2PM

https://electricplayhouse.com/

CREATE A NEW PROJECT
“Week7”

CREATE 2 CLASSES:
MyFrame.java and MyPanel.java

COPY STARTER CODE FROM CLASS SCHEDULE

TODAY:
KEYBOARD INPUT

USING KeyListener

IN MyPanel

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.KeyListener;

public class MyPanel extends JPanel implements KeyListener {
```

tells compiler you'll be using keyboard input
you'll be "listening" for key presses

ADD 3 METHODS

```
@Override
public void keyTyped(KeyEvent e) {

}

@Override
public void keyPressed(KeyEvent e) {

}

@Override
public void keyReleased(KeyEvent e) {

}
```

code that will run when a letter key is typed

code that will run when any key is typed

code that will run when any key is released

note: I will not expect you to remember these on any exam or quiz

COMPILE & RUN TO CHECK CODE

EVENT TRIGGERED CODE

- Key pressed on keyboard = an event
- The event triggers an interruption
- The program stops what it was doing
- Executes the event triggered code
- Goes back to what it was doing

ADD A VARIABLE FOR KEY INFORMATION

```
public class MyPanel extends JPanel implements KeyListener {
    int width;
    int height;
    char keyPressed;
```

will store the character that is typed

EDIT THE keyTyped METHOD

```
@Override
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
}
```

store the typed key in the keyPressed variable
print out the key that was typed

TELL THE PANEL TO LISTEN TO THE KEYBOARD

```
MyPanel(int w, int h) {
    width = w;
    height = h;
    Dimension d = new Dimension(width, height);
    setPreferredSize(d);
    setVisible(true);
    addKeyListener(this);
}
```

tells the panel to listen for keyboard input

TELL THE PANEL TO LISTEN TO THE KEYBOARD

```
MyPanel(int w, int h) {
    width = w;
    height = h;
    Dimension d = new Dimension(width, height);
    setPreferredSize(d);
    setVisible(true);
    addKeyListener(this);
    setFocusable(true);
    requestFocusInWindow();
}
```

requests keyboard focus in the panel

keyboard focus = which window you're typing in

COMPILE & RUN
SHOULD SEE KEYS PRINTING

```
f
d
g
k
h
i
l

6
7
```

questions?

DRAW THE KEY ON SCREEN

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.WHITE);
    Font f = new Font("Courier", Font.BOLD, 100);
    g.setFont(f);
}
```

create a font and
set it to be current font

DRAW THE KEY ON SCREEN

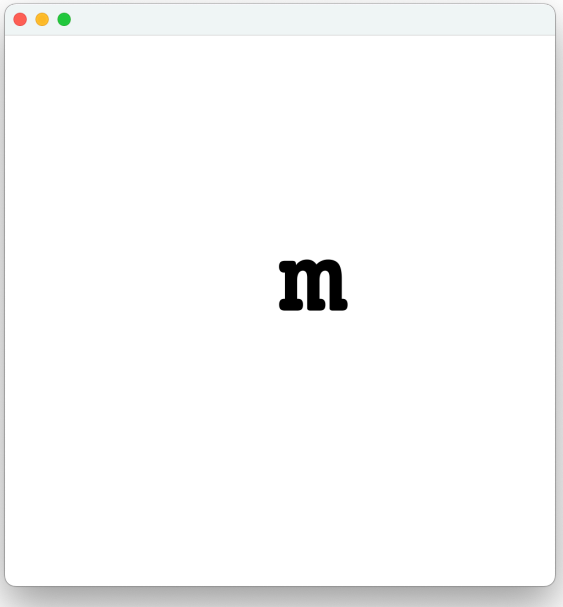
```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.WHITE);
    Font f = new Font("Courier", Font.BOLD, 100);
    g.setFont(f);
    String s = Character.toString(keyPressed);
}
```

generate a string from the keyPressed character

DRAW THE KEY ON SCREEN

```
protected void paintComponent(Graphics g) {
     super.paintComponent(g);
     setBackground(Color.WHITE);
     Font f = new Font("Courier", Font.BOLD, 100);
     g.setFont(f);
     String s = Character.toString(keyPressed);
     g.drawString(s, width/2, height/2);
}
```

draw the string in the center of the screen



THINGS TO NOTE

- All letter keys generate a character
- Can do upper case letters
- Space bar generates a character ' ‘
- Number keys and symbol keys generate characters
- Some keys do not generate characters: command, option, arrow keys, tab, etc.

questions?

CHANGE BACKGROUND TO A RANDOM PASTEL COLOR IN RESPONSE TO KEY ‘c’

ADD A VARIABLE FOR THE BACKGROUND COLOR

```
public class MyPanel extends JPanel implements KeyListener {
    int width;
    int height;
    char keyPressed;
    Color background;

    MyPanel(int w, int h) {
        width = w;
        height = h;
        background = Color.WHITE;
        Dimension d = new Dimension(width, height);
        setPreferredSize(d);
        setVisible(true);
        addKeyListener(this);
        setFocusable(true);
        requestFocusInWindow();
    }
```

USE THE VARIABLE FOR BACKGROUND COLOR

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(background);
    Font f = new Font("Courier", Font.BOLD, 100);
    g.setFont(f);
    String s = Character.toString(keyPressed);
    g.drawString(s,width/2, height/2);
}
```

CHANGE BACKGROUND WHEN ‘c’ IS PRESSED

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        //set background variable to new value
    }
}
```

CHANGING TO A RANDOM COLOR

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red =(int)(Math.random()*256);
        int green =(int)(Math.random()*256);
        int blue =(int)(Math.random()*256);
        background = new Color(red, green, blue);
    }
}
```

HOW DO WE GENERATE ONLY PASTEL COLORS?

A RANDOM PASTEL COLOR

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red = 200 + (int)(Math.random()*56);
        int green = 200 + (int)(Math.random()*56);
        int blue = 200 + (int)(Math.random()*56);
        background = new Color(red, green, blue);
    }
}
```

The minimum value for any color component is 200. Ensures colors will be light.

questions?

GENERAL CODE STRUCTURE: RESPONSE TO KEYBOARD

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red = 200 + (int)(Math.random()*56);
        int green = 200 + (int)(Math.random()*56);
        int blue = 200 + (int)(Math.random()*56);
        background = new Color(red, green, blue);
    }
}
```

Condition: which key is pressed?

GENERAL CODE STRUCTURE: RESPONSE TO KEYBOARD

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red = 200 + (int)(Math.random()*56);
        int green = 200 + (int)(Math.random()*56);
        int blue = 200 + (int)(Math.random()*56);
        background = new Color(red, green, blue);
    }
}
```

What to do if that key is pressed

CHANGE TEXT COLOR TO RANDOM DARK COLOR IF ’t’ IS PRESSED

ADD A VARIABLE FOR THE TEXT COLOR

```
public class MyPanel extends JPanel implements KeyListener {
    int width;
    int height;
    char keyPressed;
    Color background;
    Color textColor;

    MyPanel(int w, int h) {
        width = w;
        height = h;
        background = Color.WHITE;
        textColor = Color.BLACK;
        Dimension d = new Dimension(width, height);
        setPreferredSize(d);
        setVisible(true);
        addKeyListener(this);
        setFocusable(true);
        requestFocusInWindow();
    }
```

USE THE VARIABLE TO COLOR TEXT

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(background);
    Font f = new Font("Courier", Font.BOLD, 100);
    g.setFont(f);
    String s = Character.toString(keyPressed);
    g.setColor(textColor);
    g.drawString(s,width/2, height/2);
}
```

CODE STRUCTURE: RESPONSE TO KEYBOARD

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red = 200 + (int)(Math.random()*56);
        int green = 200 + (int)(Math.random()*56);
        int blue = 200 + (int)(Math.random()*56);
        background = new Color(red, green, blue);
    }
    else if (keyPressed=='t') {
        int red = (int)(Math.random()*200);
        int green = (int)(Math.random()*200);
        int blue = (int)(Math.random()*200);
        textColor = new Color(red, green, blue);
    }
}
```

Condition: which key is pressed?

CODE STRUCTURE: RESPONSE TO KEYBOARD

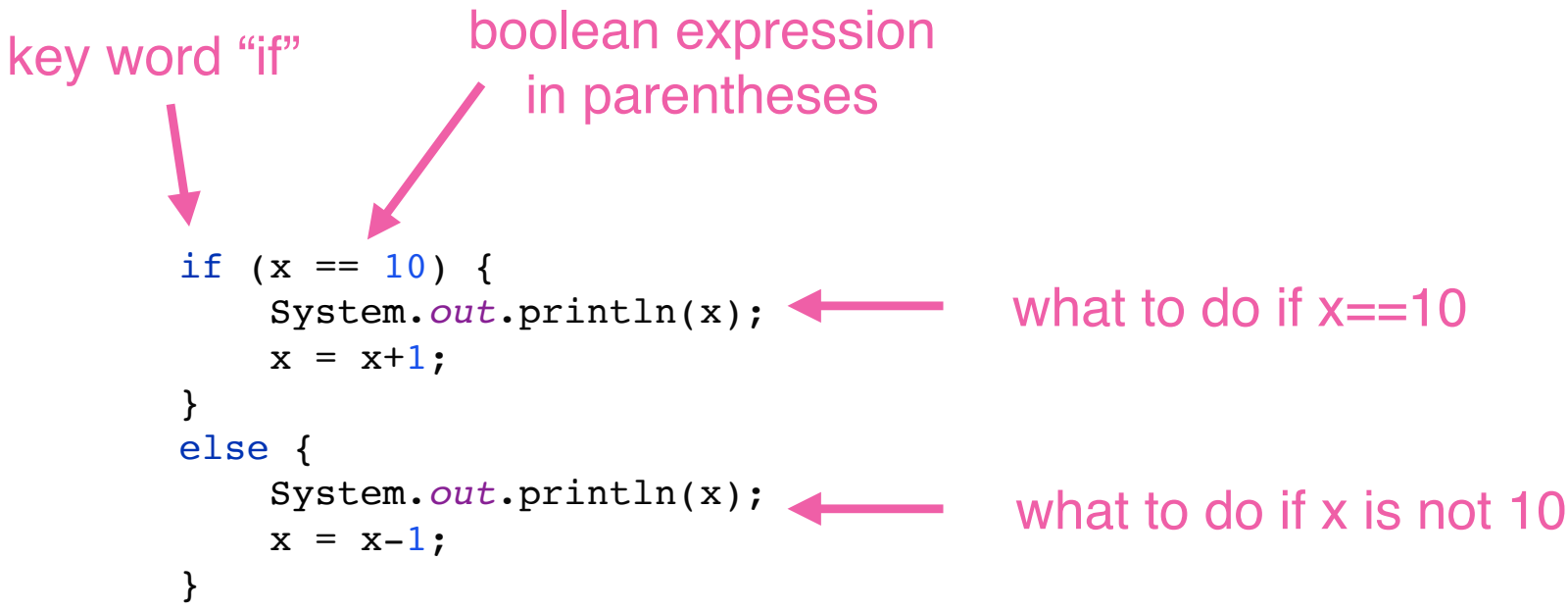
```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red = 200 + (int)(Math.random()*56);
        int green = 200 + (int)(Math.random()*56);
        int blue = 200 + (int)(Math.random()*56);
        background = new Color(red, green, blue);
    }
    else if (keyPressed=='t') {
        int red = (int)(Math.random()*200);
        int green = (int)(Math.random()*200);
        int blue = (int)(Math.random()*200);
        textColor = new Color(red, green, blue);
    }
}
```

What to do if that key is pressed

questions?

A DIFFERENT STRUCTURE: SWITCH STATEMENTS

IF ELSE STATMENT



MANY IF ELSE STATMENTS

```
if (x == 10) {
    System.out.println(x);
    x = x+1;
}
else if (x == 20) {
    System.out.println(x);
    x = x-1;
}
else if (x == 30) {
    System.out.println(x);
    x = x-2;
}
else {
    System.out.println(x);
    x = x-5;
}
```

what to do if x == 10

what to do if x == 20

what to do if x == 30

what to do if none of the other conditions are true

A SWITCH STATEMENT

```
switch (x) {
    case (10):
        System.out.println(x);
        x = x+1;
        break;
    case (20):
        System.out.println(x);
        x = x-1;
        break;
    case (30):
        System.out.println(x);
        x = x-2;
        break;
    default:
        System.out.println(x);
        x = x-5;
        break;
}
```

x is variable to test

means (x==10)

means (x==20)

means (x==30)

similar to final else

A SWITCH STATEMENT

```
switch (x) {
    case (10):
        System.out.println(x);
        x = x+1;
        break;
    case (20):
        System.out.println(x);
        x = x-1;
        break;
    case (30):
        System.out.println(x);
        x = x-2;
        break;
    default:
        System.out.println(x);
        x = x-5;
        break;
}
```

what to do if x == 10

what to do if x == 20

what to do if x == 30

what to do if none of the other conditions are true

CODE WITH IF/ELSE STATEMENTS

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    if (keyPressed=='c') {
        int red = 200 + (int)(Math.random()*56);
        int green = 200 + (int)(Math.random()*56);
        int blue = 200 + (int)(Math.random()*56);
        background = new Color(red, green, blue);
    }
    else if (keyPressed=='t') {
        int red = (int)(Math.random()*200);
        int green = (int)(Math.random()*200);
        int blue = (int)(Math.random()*200);
        textColor = new Color(red, green, blue);
    }
}
```

CODE WITH SWITCH STATEMENT

```
public void keyTyped(KeyEvent e) {
    keyPressed = e.getKeyChar();
    System.out.println(keyPressed);
    switch (keyPressed) {
        case('c'):
            int red = 200 + (int)(Math.random()*56);
            int green = 200 + (int)(Math.random()*56);
            int blue = 200 + (int)(Math.random()*56);
            background = new Color(red, green, blue);
            break;
        case('t'):
            int red = (int)(Math.random()*200);
            int green = (int)(Math.random()*200);
            int blue = (int)(Math.random()*200);
            textColor = new Color(red, green, blue);
            break;
    }
}
```

questions?

SPECIAL GUESTS

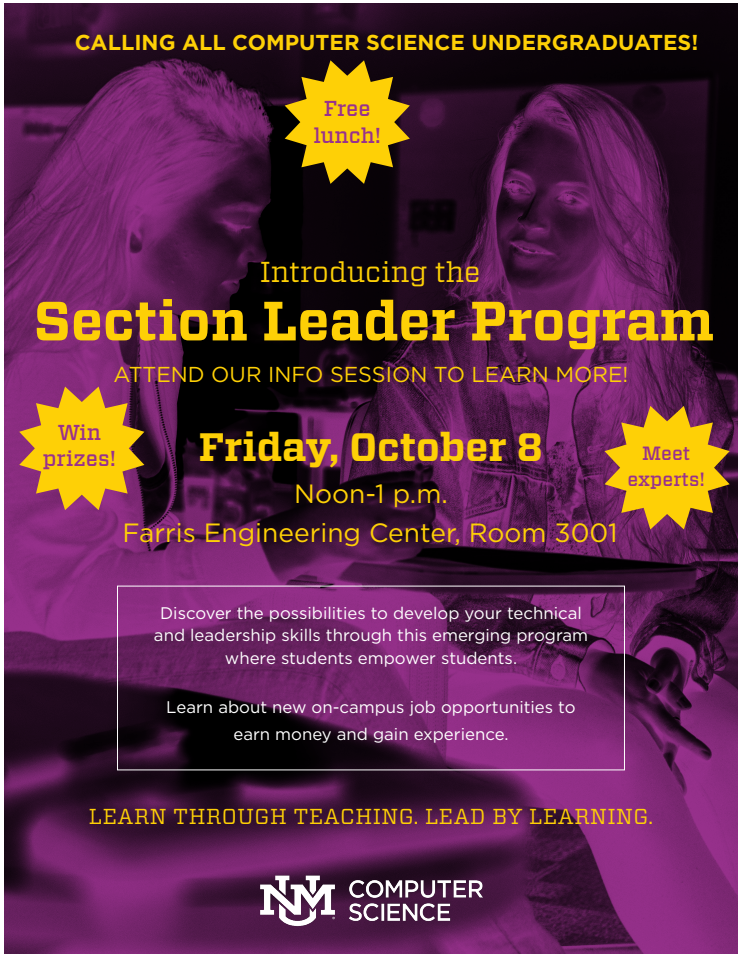
Dr. Polle Zellweger earned a BA in Math from the U of Alaska, **an MS in Computing & Information Science from UNM**, and a PhD in Computer Science from UC Berkeley. She spent 20 years as a user interface researcher at Xerox PARC, where she explored digital audio and video applications, electronic documents, hypermedia, and information visualization. She has also taught at the U of Aarhus in Denmark and the U of Washington. She is currently an independent consultant in Seattle. **She was honored with a Centennial Distinguished Alumni Award by the UNM College of Engineering in 1989.**

Dr. Jock Mackinlay earned a BS in Math and Computer Science from UC Berkeley and a PhD in Computer Science from Stanford. He spent 20 years as a user interface researcher at Xerox PARC, where he pioneered work on information visualization. He then joined Tableau Software (now part of Salesforce) in Seattle to help people see and understand data, where he started and led their research team and their user experience team. He is currently a Technical Fellow at Tableau/Salesforce.

Juan Benet earned a BS in Computer Science from Stanford in 2010. He is the founder of Protocol Labs, a computer networks R&D startup that is currently working on Web 3.0. Their current focus is Filecoin, a cryptotoken-incentivized decentralized file storage network.

Molly Mackinlay earned a BS in Computer Science from Stanford in 2013. She was a product manager at Google for five years in Mountain View and New York City, where she worked on education tools, Chrome, and search. She now leads the product and engineering teams at Protocol Labs.

COME TO LUNCH ON FRIDAY!



Thank you!

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/