CLASS 6: FEB 11
OUTLINE

• Debrief Homework 1
• Revisit Honor Code
• Demo Eclipse debugging
• Revisit Interfaces
• Graphical User Interfaces (GUIs)
DEBRIEF HOMEWORK 1
FROM REFLECTIONS

• Java is more different than Python than I expected
• I learned a lot from this assignment
• Too specific or too open ended?
• Several comparisons to natural languages (substitutions)
• Integrating and synthesizing many concepts already:
  • Working with strings
  • For loops
  • If/else statements and boolean logic
  • User input
  • Arrays
  • Printing complex objects
Main takeaway:

trajectory/improvement is valued,
don’t worry if homework 1 didn’t go well

Labs:

3: fully complete
2: almost there
1: submitted something functional
0: no submission or non-functional
Homeworks:

reflection: 1pt

typescript or screenshot: 1pt

progress toward a functional solution (including style): 5pts

3 key concepts successfully executed: 1pt each
  * outer loop of interaction between user and computer
  * correct mirroring of words with inner loop
  * correct use of arrays (allocation, indexing, printing)
SUGGESTIONS FOR FUTURE HOMEWORKS

• Start early!
• Try to maintain a state of having something “working”
• Use Javadocs as much as possible
  • We won’t cover every method you’ll need for homework
• Just try it! (from the TAs)
SUGGESTIONS FOR FUTURE HOMEWORKS

• Start early!
• Try to maintain a state of having something “working”
• Use Javadocs as much as possible
  • We won’t cover every method you’ll need for homework
• Just try it! (from the TAs)

• Class size issues (key: make sure you can rsubmit early!)
  • Late work or work not submitted over the server
  • No typescript/screenshot
  • Filenames and file formats need to be correct
HONOR CODE
Main takeaway:

Think of the honor code as a collaboration, we’re figuring out what original work means for this class
Main takeaway:

Think of the honor code as a collaboration, we’re figuring out what original work means for this class

Okay:

Google: “java what is the structure of a for loop”
“concatenate strings in java”
“java out of bounds error”
HONOR CODE

Main takeaway:
Think of the honor code as a collaboration, we’re figuring out what original work means for this class

Okay:
Google: “java what is the structure of a for loop”
“concatenate strings in java”
“java out of bounds error”

Not okay:
“Hi friend who took 212 before, can I see your code?”
Google: “java chatbot code”
HONOR CODE

Main takeaway:

Think of the honor code as a collaboration, we’re figuring out what original work means for this class

Okay:

Google: “java what is the structure of a for loop”
“concatenate strings in java”
“java out of bounds error”

Not okay:

“Hi friend who took 212 before, can I see your code?”
Google: “java chatbot code”
HONOR CODE

• If you were alone in a room with no internet and no notes, could you reproduce your submission?
  (starting from any code given for the assignment)

• Do you understand every line of code that you wrote?
  • There are a lot of built-in classes in Java, but make sure you know what they are doing (Javadoc).
ECLIPSE DEBUGGING DEMO
GRAPHICAL USER INTERFACES
USER INTERFACES

• You use these everyday!
• Terminal
• Websites
• Games
• Apps
• Coffee maker buttons

• **User Interface** is a very broad term: facilitating the interaction between human and machine

• GUI (Graphical User Interface), pronounced “gooey”
• Subset of UI (User Interface), pronounced “you I”
EXAMPLE: FOLDER/FILE NAVIGATION

Wikipedia quote:
“GUls were introduced in reaction to the perceived steep learning curve of command-line interfaces.” (i.e. the terminal)
EXAMPLE: FOLDER/FILE NAVIGATION

TEXT-BASED USER INTERFACE

GRAPHICAL USER INTERFACE

Wikipedia quote:
“GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces.” (i.e. the terminal)
LAB 3 (TODAY)

https://docs.oracle.com/javase/7/docs/api/java/awt/event/ActionListener.html
LAB 3 (TODAY)

https://docs.oracle.com/javase/7/docs/api/java/awt/event/ActionListener.html

```java
/** Event handler for Cycle button */
private class CycleListener implements ActionListener {

    /**
     * Cycles the colors when the button is pushed.
     * @param e Holds information about the button-push event
     */
    public void actionPerformed(ActionEvent e) {
        bullseye.cycle(); // note reference to enclosing class's private field
    }
}
```
LAB 3 (TODAY)

Add new action listener to the button:

```java
JButton cycleButton = new JButton("Cycle");
panel.add(cycleButton);
cycleButton.addActionListener(new CycleListener());
```