



# **COMP 110/401**

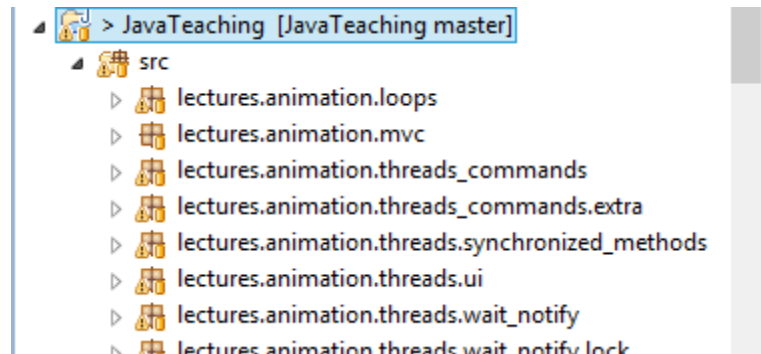
## **APPENDIX: SEARCHING FOR TEXT IN ECLIPSE**

**Instructor: Prasan Dewan (FB 150, [dewan@unc.edu](mailto:dewan@unc.edu))**

# FILE SEARCH IN A PROJECT (USEFUL IN JAVA TEACHING AND YOUR PROJECTS)

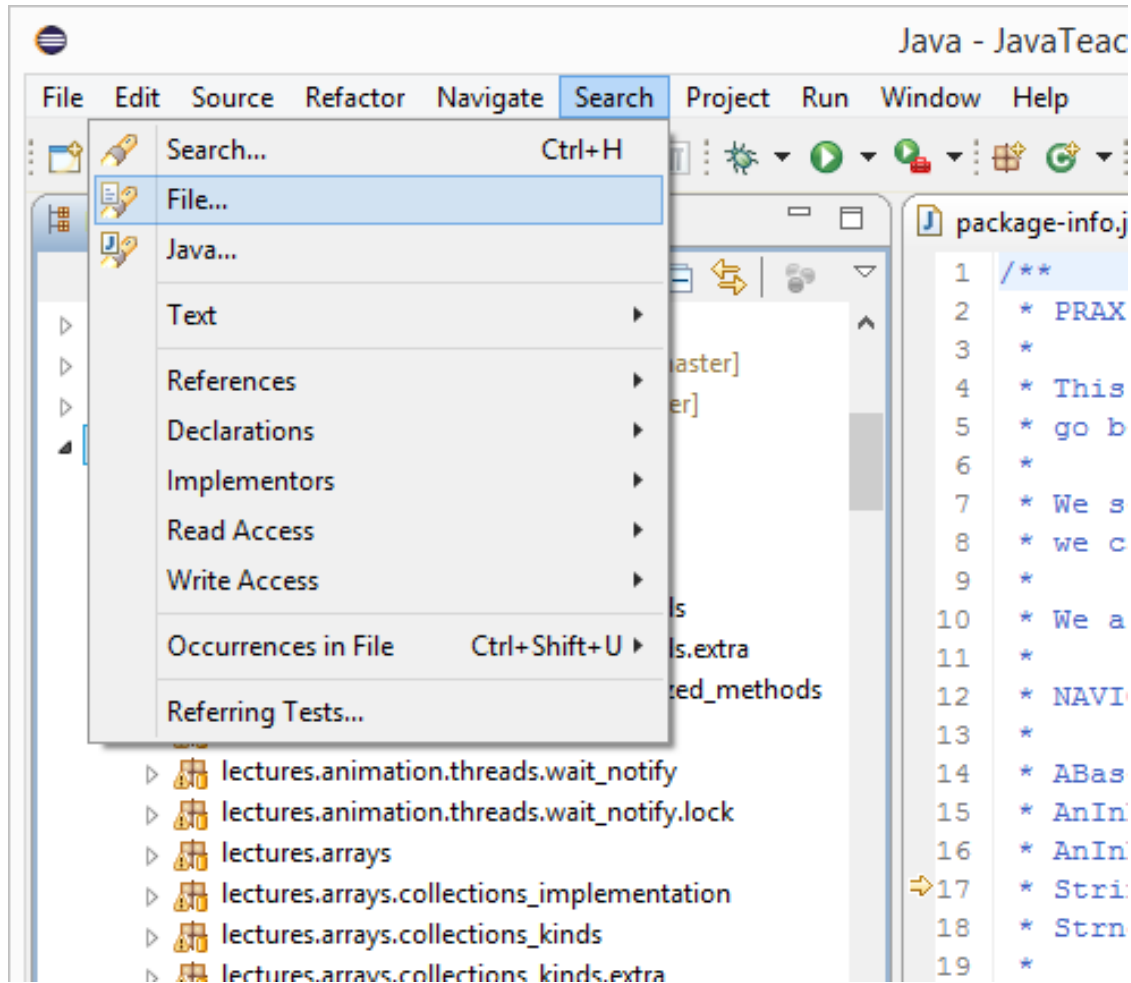
Suppose you want to search the entire project for some string such as some text in a Sakai question: such as “println(c) displays”

# SELECT PROJECT



Select project (not the src folder)

# FILE → SEARCH



# SEARCH DIALOG

The screenshot shows the Eclipse Search dialog box. It has a title bar with standard window controls. Below the title bar is a tabbed interface with five tabs: 'File Search' (selected), 'Task Search', 'Git Search', 'Java Search', and 'Plug-in Search'. The 'Containing text:' section has a text input field containing 'println(c) displays', which is highlighted by a blue box labeled 'Search string'. To the right of this field are three checkboxes: 'Case sensitive' (checked), 'Regular expression' (unchecked), and 'Whole word' (unchecked). Below this is a section for 'File name patterns (separated by comma):' with a dropdown menu showing '\*.java' and a 'Choose...' button. A note below the dropdown explains the wildcard symbols. There is also an unchecked checkbox for 'Consider derived resources'. The 'Scope' section has three radio buttons: 'Workspace' (unchecked), 'Selected resources' (unchecked), and 'Enclosing projects' (checked), which is highlighted by a blue box labeled 'Scope of search'. Below these are a 'Working set:' label, an empty text field, and another 'Choose...' button. At the bottom of the dialog are four buttons: a help button (question mark icon), 'Customize...', 'Replace...', and 'Search', which is highlighted by a blue box labeled 'Command'.

Search

File Search Task Search Git Search Java Search Plug-in Search

Containing text:

println(c) displays

Search string

☒ Case sensitive  
☐ Regular expression  
☐ Whole word

(\* = any string, ? = any character, \ = escape for literals: \* ? \)

File name patterns (separated by comma):

\*.java

Choose...

(\* = any string, ? = any character, \x = excluding x)

☐ Consider derived resources

Scope

☐ Workspace ☐ Selected resources ☒ Enclosing projects

Scope of search

☐ Working set: Choose...

? Customize... Replace... Search

Command

# SEARCH RESULTS

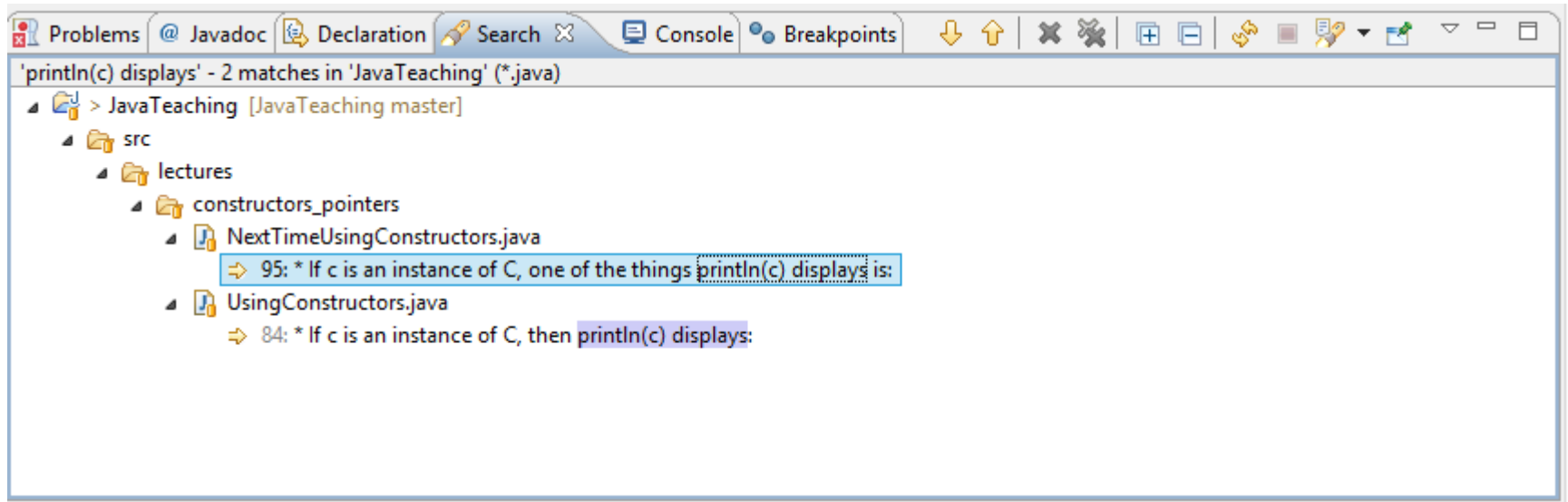
The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Displays the project structure for 'grail13', including packages like 'lectures.animation.loops', 'lectures.animation.mvc', and 'lectures.class\_dual\_roles.instances'.
- Code Editor:** Shows the file 'package-info.java' with the following content:

```
1 /**
2  * PRAXIS INFO
3  *
4  * This praxis teaches inheritance using the example of collections that
5  * go beyond fixed-length arrays.
6  *
7  * We see three such collections- histories, databases, and sets- and see how
8  * we can reuse (share) code among them using inheritance.
9  *
10 * We also see factory methods that choose between different histories.
11 *
12 * NAVIGATION INFO
13 *
14 * ABaseStringHistory
15 * AnInheritingStringDatabase
16 * AnInheritingStringSet
17 * StringHistoryFactory
18 * StrngHistoryReverser
19 *
20 */
21 package lectures.inheritance;
```
- Search View:** Shows the search results for the query 'println(c) displays'. The results are listed under the 'JavaTeaching' project, specifically in the 'lectures' package, under 'constructors\_pointers'. The results are:
  - 95: \* If c is an instance of C, one of the things `println(c)` displays
  - 84: \* If c is an instance of C, then `println(c)` displays:
- Console:** Shows the output of the search results, including the text 'println(c) displays'.

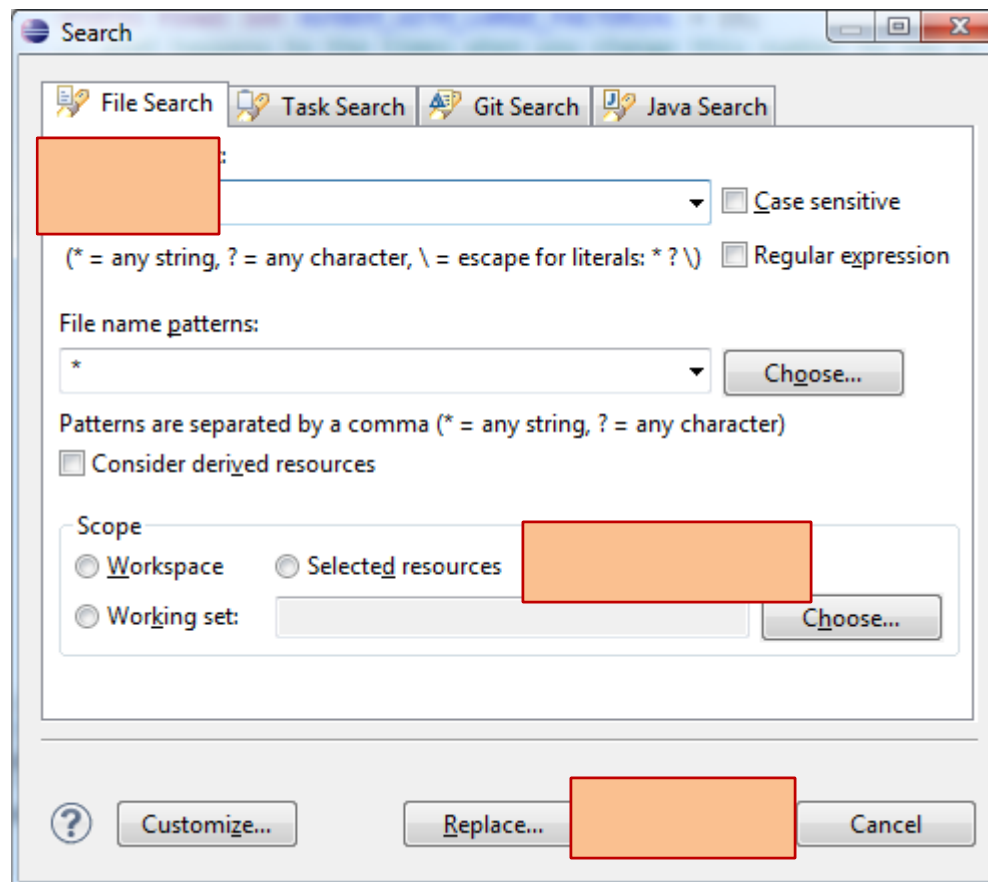
Search view shown,  
console in another tab

# SEARCH RESULTS ZOOMED



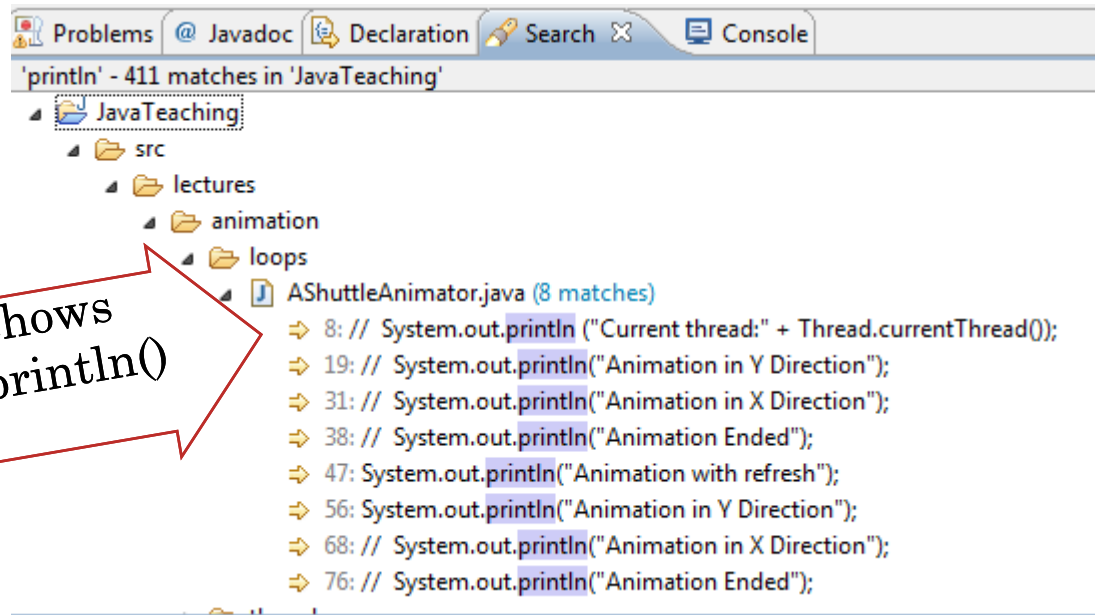
Look at all the tabs  
availalale here

# MULTI FILE TEXT SEARCH FOR PRINTLN





# SEARCH RESULTS



Use CTR+SHIFT+G (shown earlier) to find active uses of the method