Debugging an Applet Using Appletviewer

- Compile your applet using javac, making sure you add the –g flag. Without –g, you won't be able to list the file or inspect variables.
- Start **appletviewer** in "-debug" mode, specifying the HTML file, NOT the applet to debug.
    - **jdb** will start, and report *Initializing jdb …*
- Enter the **use** command. This is critical. Since **jdb** runs in its own environment, it will NOT use your classpath.
    - **use** <path to codebase/documentbase>
- Enter **run** to start the AppletViewer (might take a few seconds).
    - **jdb** will report *run sun.applet.Main <html file from command line>*
    - The applet will appear.
- Set a breakpoint. The commands are stop in and stop on, and they set breakpoints by function name and line number, respectively. Examples (note use of period and colon):
    - Stop in Test.actionPerformed
    - Stop at Test:34
- Carry out an action to cause the breakpoint to be reached (click a button?). **jdb** will stop the applet's execution at the breakpoint just set.
- Issue the command **list** to view the source code where the program stopped.
- Use the *step into* command **step** to execute the next line. If it is a function call, the function is entered. **next** is the *step over* command.
- **up** will move you to the the line of the calling function where the present function was called (only do this if the method you stopped in called the method you are in).
- **down** will reverse the effect of **up**.
- To see the local variables, enter **locals**, or the name of the variable (use this to print the object whose method you are stopped in).
- **cont** will continue execution from the breakpoint. The breakpoint message shows the line number of the breakpoint. To clear that breakpoint use **clear <Class name> <line number>**.
- Use the **print** command to inspect variables. Locals need no prefix. **print** *this will output the object.
- See other documents for more advanced commands.