

Hands-On: Collections

This activity is designed to illustrate the design and utility of elements of the Java Collections Framework (JCF). You should study the [instructional resources](#) on Collections before attempting this activity.

You will need the following source code files to complete this activity.

- [Song.java](#)
- [Playlist.java](#)

Note: This activity utilizes [JGRASP Viewers](#), which are available in [JGRASP](#), [IntelliJ](#), and [Eclipse](#).

Playlist

1. Open `Playlist.java` , then compile and run it. Observe the output to understand what the `main` method is doing.
2. Set a breakpoint on the following line in the `main` method: `List<Song> playlist = new ArrayList<>();`
3. Start the debugger and wait until execution is paused at the breakpoint.
4. Step over this statement and then raise a viewer on the `playlist` object.
5. Step over the next several statements and watch the viewer as it shows the effect of adding songs to the `playlist` .
6. Note that four different implementing classes of the `List` interface are already imported (`ArrayList` , `LinkedList` , `Stack` , and `Vector`). Systematically change the instantiation of `playlist` to each of these classes in turn. Run and debug the program each time to observe that the functionality of the program doesn't change although the exact nature of the `playlist` object does indeed change.

Working with collections

1. Practice working with collections by completing the `runningTime` method in the `Playlist` class. Note that your solution will work for any class that implements the `List` interface.

Submission

The submission page for this activity asks you to complete the `runningTime` method above and then submit it for a grade.