Activity 7.1

The class was shown this code,

```
public void engage1() {
  String artist = "Wynton Marsalis";
  Song s1 = new Song("lwvaRkyT2A8",artist,"Hot House Flowers");
  Song s2 = new Song("GZt0p1DcbyY",artist,"Lazy Afternoon");
  Song s3 = new Song("EB_BO_33JAw","Anna Ternheim","No, I don't
Remember");
  // DRAW OBJECT DIAGRAM SHOWING STATE OF PROGRAM HERE
}
public class Song{
    private ArrayList<Integer> ratings;
    private String youtubeID;
    private String artist;
    private String title;
    public Song(String youtubeID, String artist, String title){
        this.youtubeID = youtubeID;
        this.artist = artist;
        this.title = title;
        this.ratings = new ArrayList<>();
    }
      // The rest of the class is omitted
}
```

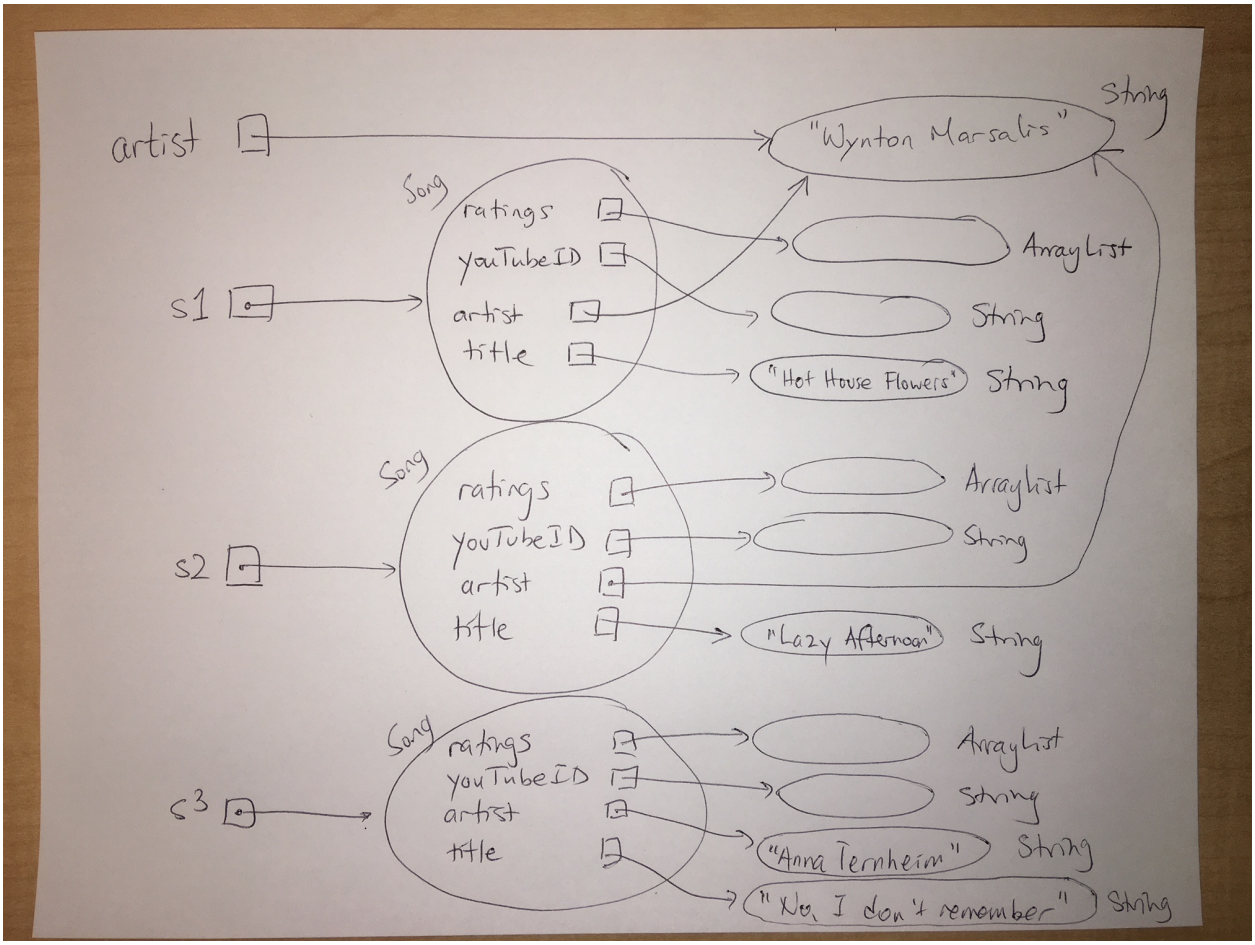
and was given these questions to discuss:

Q1: Draw an object diagram showing the state of the program after s3 is initialized.
Q2: How many objects exist in memory?
Q3: How many variables exist in memory?
Q4: How many instance variables exist in memory?

Consider ONLY those variables, classes that are explicitly mentioned in the code. For example, depict the ArrayList in your diagram, but do NOT depict the internals of the ArrayList since the definition of the ArrayList class is not given.

(answers on next page)

Q1



Q2 How many objects exist in memory?

Count the ovals in the diagram: 14.

Q3 How many variables exist in memory?

Count the boxes in the diagram: 16.

Q4 How many instance variables exist in memory?

Count the boxes inside ovals: 12.

Assessment 1

The class was shown this code,

```
public void assess1() {
Pair p1 = new Pair("Leonard Cohen",
                   new JLabel("Singer/Songwriter"));
Pair p2 = new Pair("Leonardo da Vinci",
                   new JLabel("Inventor/Artist/etc"));
// CONSIDER STATE OF PROGRAM HERE
}
public class Pair {
     private String first;
     private JLabel second;
     public Pair(String s, JLabel jl) {
          first = s;
          second = jl;
     }
     public String getString() { return first; }
     public JLabel getLabel() { return second; }
}
```

And was told,

> **Consider ONLY those variables, classes that are explicitly mentioned in the code. For example, depict the JLabel in your diagram, but do NOT depict the internals of the JLabel since the definition of the JLabel class is not given.**

The class had to answer these questions:

> **Q1: How many objects exist in memory?**
> **Q2: How many instance variables exist in memory?**
>
> **HINT: Draw an object diagram showing the state of the program after p2 is initialized.**

Q1: 8 (2 Pair, 4 String, 2 JLabel)
Q2: 4 (2 String first, 2 JLabel second)

Activity 7.2

The class was shown this code,

```
public void engage() {
  String artist = "Wynton Marsalis";
  Song s1 = new Song("lwvaRkyT2A8",artist,"Hot House Flowers");
  Song s2 = new Song("GZt0p1DcbyY",artist,"Lazy Afternoon");
  Song s3 = new Song("EB_BO_33JAw","Anna Ternheim","No, I don't
Remember");
  // Add code here to call displaySongsInWindows method to
  // display these three songs
}

public void displaySongsInWindows(ArrayList<Song> tunes) {
  // Define this method so that for each Song in tunes it
  // creates and displays a new JFrame, whose title is the
  // name of the artist, and which contains a JLabel
  // containing the song's title.
}
```

A possible solution is shown below. For the engage method:

displaySongsInWindows expects an ArrayList of Song objects as an argument, so we must build that ArrayList and populate it with our Songs, prior to the method call. We must also then supply that ArrayList as an argument in the method call"

```
public void engage() {
  String artist = "Wynton Marsalis";
  Song s1 = new Song("lwvaRkyT2A8",artist,"Hot House Flowers");
  Song s2 = new Song("GZt0p1DcbyY",artist,"Lazy Afternoon");
  Song s3 = new Song("EB_BO_33JAw","Anna Ternheim","No, I don't
Remember");
  ArrayList<Song> myTunes = new ArrayList<Song>();
  myTunes.add(s1);
  myTunes.add(s2);
  myTunes.add(s3);
  displaySongsInWindows(myTunes);
}
```

For the displaySongsInWindows methods:

Loop through the Song objects in the parameter, tunes.
For each Song:
  create JFrame and set its title to the artist of the Song,
  create a JLabel and set its text to the title of the Song,

add the JLabel to the JFrame, then pack the JFrame and make it visible.
The finished result is:

```
public void displaySongsInWindows(ArrayList<Song> tunes) {
  for (Song s : tunes) {
    JFrame w = new JFrame(s.getArtist());
    JLabel jl = new JLabel(s.getTitle());
    w.add(jl);
    w.pack();
    w.setVisible(true);
  }
}
```

Assessment 2

The class was shown this information,

```
public void displaySongsInWindows(ArrayList<Song> tunes) {
  for (Song s : tunes) {
    JFrame w = new JFrame(s.getArtist());
    JLabel jl = new JLabel(s.getTitle());
    w.add(jl);
    w.pack();
    w.setVisible(true);
  }
}
```

Q1: How many objects will be created by displaySongsInWindows when it is called? Assume that tunes is not null, and consider only objects that are instantiated directly in this method.

The loop body creates one new JFrame and one new JLabel for each loop iteration. Since we don't know how large tunes is all we can say is tunes.size() * 2

Q2: Suppose the following code is run. How many objects will be created by displaySongsInWindows when it is called? (i.e. ignore the Song and ArrayList objects).

```
Song s = new Song("EB_BO_33JAw","Anna Ternheim","No, I
don't Remember");
ArrayList<Song> myTunes = new ArrayList<Song>();
myTunes.add(s);
myTunes.add(s);
myTunes.add(s);
displaySongsInWindows(myTunes);
```

In this case, since myTunes has size 3, the total number of objects created will be tunes.size() * 2 = 3 * 2 = 6.