Playing with Objects (example)

```java
public class CD {
    private String title;
    private String artist;
    private String[] songTitles;
    private int releaseYear;

    CD(String theTitle, String theArtist, int theReleaseYear, int num) {
        this.title = theTitle;
        this.artist = theArtist;
        this.releaseYear = theReleaseYear;
        this.songTitles = new String[num];
    }
}
```

Define Constructor Method for Initialization
Playing with Objects (example)

<table>
<thead>
<tr>
<th>CD</th>
<th>public class CD {</th>
</tr>
</thead>
<tbody>
<tr>
<td>String title</td>
<td>private String title;</td>
</tr>
<tr>
<td>String artist</td>
<td>private String artist;</td>
</tr>
<tr>
<td>String[] songTitles</td>
<td>private String[] songTitles;</td>
</tr>
<tr>
<td>int releaseYear</td>
<td>private int releaseYear;</td>
</tr>
<tr>
<td>addSong(String):boolean</td>
<td>public boolean addSong(String theSong){</td>
</tr>
<tr>
<td>hasDuplicates(String):boolean</td>
<td>//pre: String representing the song</td>
</tr>
<tr>
<td>showInfo():void</td>
<td>// to be added</td>
</tr>
<tr>
<td>prettyPrintSongs():void</td>
<td>//post: TRUE if addition is succesful,</td>
</tr>
<tr>
<td></td>
<td>// FALSE otherwise. Do not allow</td>
</tr>
<tr>
<td></td>
<td>// duplicates (ignore case)</td>
</tr>
<tr>
<td></td>
<td>if (!hasDuplicates(theSong)){</td>
</tr>
<tr>
<td></td>
<td>return addThisSong(theSong);</td>
</tr>
<tr>
<td></td>
<td>}else{</td>
</tr>
<tr>
<td></td>
<td>return false;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
//Instance Methods
public boolean addSong(String theSong) {
    //pre: String representing the song to be added
    //post: TRUE if addition is successful, FALSE otherwise.
    // Do not allow duplicates (ignore case)
    if (!hasDuplicates(theSong)) {
        return addThisSong(theSong);
    } else {
        return false;
    }
}

public boolean hasDuplicates(String theSong) {
    // pre: String with the song title
    // post: TRUE if name exists in songTitles, FALSE otherwise
    for (int i = 0; i < songTitles.length && songTitles[i] != null; i++) {
        if (songTitles[i].equalsIgnoreCase(theSong)) {
            return true;
        }
    }
    return false;
}
Playing with Objects (example)

```java
// Instance Methods
public boolean addSong(String theSong){
    // pre: String representing the song to be added
    // post: TRUE if addition is successful, FALSE otherwise.
    // Do not allow duplicates (ignore case)
    if (!hasDuplicates(theSong)){
        return addThisSong(theSong);
    } else {
        return false;
    }
}

private boolean addThisSong(String theSong){
    // pre: String with the song title
    // post: TRUE if addition succeeds, FALSE otherwise
    int i = 0;
    for(; i < songTitles.length && songTitles[i] != null; i++){
        ;
    }
    if (i < songTitles.length){
        songTitles[i] = theSong;
        return true;
    } else {
        return false;
    }
}
```
Playing with Objects (example)

```java
public void showInfo()
{
    // Pretty print CD Information
    System.out.println(" **** CD Info **** ");
    System.out.println(" CD Title :	"+ title+"\n Artist :	"+artist+
                        +"\n Year :	"+releaseYear);
    System.out.println("\tSong List :");
    prettyPrintSongs();
    System.out.println(" \t\t\t\t**** CD Info **** ");
}

public void prettyPrintSongs()
{
    //Pretty print Song List
    for(int i = 0 ; i < songTitles.length && songTitles[i]!=null;i++){
        System.out.println("\t\t"+i+"."+songTitles[i]);
    }
}
```
public class Main{
    public static void main(String[] args){
        CD pFloyd = new CD("Wish you were here", "Pink Floyd", 1975, 5);
        pFloyd.addSong("Shine on you crazy diamond");
        pFloyd.addSong("Welcome to the machine");
        pFloyd.addSong("Have a cigar");
        pFloyd.addSong("Wish you were here");
        pFloyd.addSong("Shine on you crazy diamond (version 2)");

        CD nCave = new CD("Let Love In", "Nick Cave", 1996, 10);
        nCave.addSong("Do You Love Me?");
        nCave.addSong("Nobody's Baby Now");
        nCave.addSong("Loverman");
        nCave.addSong("Jangling Jack");
        nCave.addSong("Red Right Hand");
        nCave.addSong("I Let Love In");
        nCave.addSong("Thirsty Dog");
        nCave.addSong("Ain't Gonna Rain Anymore");
        nCave.addSong("Lay Me Low");
        nCave.addSong("Do You Love Me? (Part 2)");

        pFloyd.showInfo();
        nCave.showInfo();
        rhead.showInfo();
    }
}
## Playing with Objects (example)

- **CDCollection**
  - `CD[] theCollection`
  - `showInfo()`:void
  - `addCD(CD aCD)`

**Write `showInfo()`**

- **showInfo()**
  - show all contents of cd collection.
- **addCD(CD aCD)**
  - add a CD to the collection.
  - You should not accept duplicates. You should check
    - **artist, title, year, songs**
Todo list for Java Classes

- Define Instance Variables
  - String name;

- Define Constructor
  - Same name as class name. Use `this` to initialize your instance variables with values

- Define Instance Methods
  - follow the signature, break repetitive operations into `private` sub methods.

- In Main
  - instantiate an object of your new class and start sending messages to it.
Class variables and methods

- There are actually two entities in your programs
  - Classes and instances (objects)
- Classes can have methods and variables
  - define using the modifier static
    - static methods
    - static variables
- Static variables and methods are associated with the class rather than the instances
Defining Class variables

- `public static int maxNumber=12;`
  - one copy of maxNumber, public access and an integer

- `public static final double PI=3.14;`
  - PI is a double and final does not allow you to alter its value
  - final can be used on instance variables as well.

```java
public class Circle{
    public static int maxNumber=12;
    public static final double PI = 3.14;

    ....
}
```
Using Class Variables

- `<ClassName>..<staticVariable>`
  - gives you access to static variables.
- `public static = global variable`
  - you can access them from anywhere in the program

```java
class Circle {
    public static int maxNumber = 12;
    public static final double PI = 3.14;
    ....
}

public class Main{
    public static void main(String[] args){
        double area;
        area = Circle.PI*(2*2);
        Circle.maxNumber = 3;
    }
}
```
Defining Class Methods

- `<modifiers> static <Identifier>(<args>) {<block>}`
  - `public static double radiansToDegrees(double radians)`

- The body of a static method can
  - access other static variables or methods
  - **cannot** access instance variables or methods

```java
public class Circle{
    public static int maxNumber=12;
    public static final double PI = 3.14;

    public static double radiansToDegrees(double radians){
        return rads*180/Circle.PI;
    }
    ....
}
```
Using Class Methods

- `<ClassName>..<staticMethodName>(<argValues>)`
  - sends a message to a static method.
- `public static = global method`
  - you can access them from anywhere in the program

```java
public class Circle{
    public static int maxNumber=12;
    public static final double PI = 3.14;

    public static double radiansToDegrees(double radians){
        return rads*180/Circle.PI;
    }

    ....
}
```
# Putting it in a picture

<table>
<thead>
<tr>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>+static int maxNumber;</td>
</tr>
<tr>
<td>+static double PI;</td>
</tr>
<tr>
<td>double radius;</td>
</tr>
<tr>
<td>+getRadius():double</td>
</tr>
<tr>
<td>+setRadius(double):void</td>
</tr>
<tr>
<td>+getArea():double</td>
</tr>
<tr>
<td>+static radiansToDegrees(double):double</td>
</tr>
</tbody>
</table>

```java
public class Main {
    public static void main(String[] args) {
        Circle c1 = new Circle(3);
        Circle c2 = new Circle(6);
        c1.getArea();
        c2.getArea();
    }
}
```

<table>
<thead>
<tr>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI=3.14</td>
</tr>
<tr>
<td>maxNumber=12</td>
</tr>
<tr>
<td>radiansToDegrees(double)</td>
</tr>
</tbody>
</table>

| c1:C Circle |
| radius=3 |
| c2:C Circle |
| radius=6 |