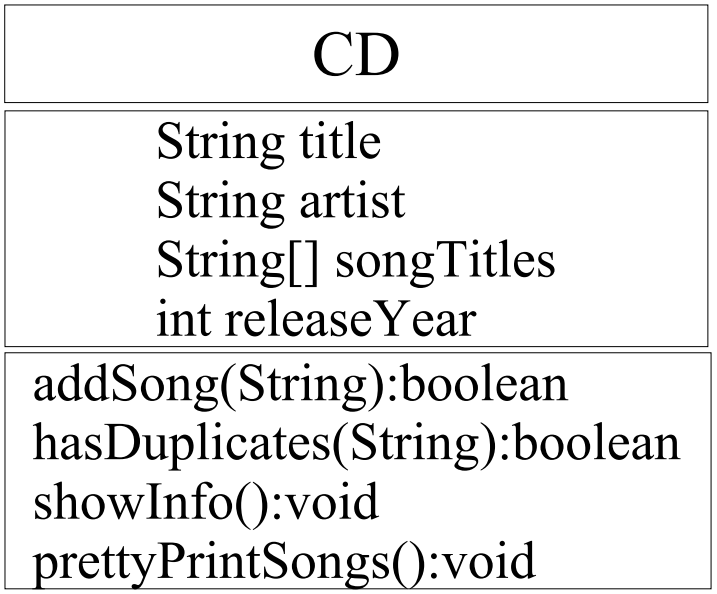
Playing with Objects (example)



Define Constructor Method for Initialization

```
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];
  }
}
```

Playing with Objects (example)

CD
String title String artist String[] songTitles int releaseYear
addSong(String):boolean hasDuplicates(String):boolean showInfo():void prettyPrintSongs():void

```
public class CD {

   private String title;
   private String artist;
   private String[] songTitles;
   private int releaseYear;



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

Playing with Objects (example)

```
//Instance Methods
public boolean addSong(String theSong){
   //pre: String representing the song to be added
   //post: TRUE if addition is succesful,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)

```
 //Instance Methods
 public boolean addSong(String theSong){
    //pre: String representing the song to be added
    //post: TRUE if addition is succesful,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)

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

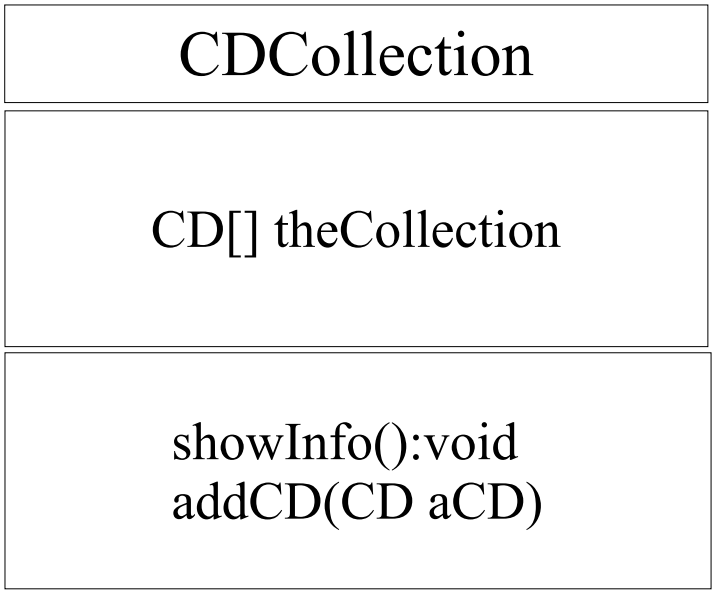
Playing with Objects (example)

```
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)



- `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

Write `showInfo()`

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.

```
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

```
public 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 radiansToDegress(double radians)`
- The body of a static method can
  - access other static variables or methods
  - cannot access instance variables or methods

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

  public static double radiansToDegress(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

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

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

....
}
```

Putting it in a picture

DIAGRAM

Circle
+static int maxNumber; +static double PI; double radius;
+getRadius():double +setRadius(double):void +getArea():double +static radiansToDegrees(double):double

CODE

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

MEMORY