CISC 124 QWIC JAVA

A Real-Life View of OOP

```
public class Dog {

//

} // end class Dog
```

Dog is a class, this is it’s object definition

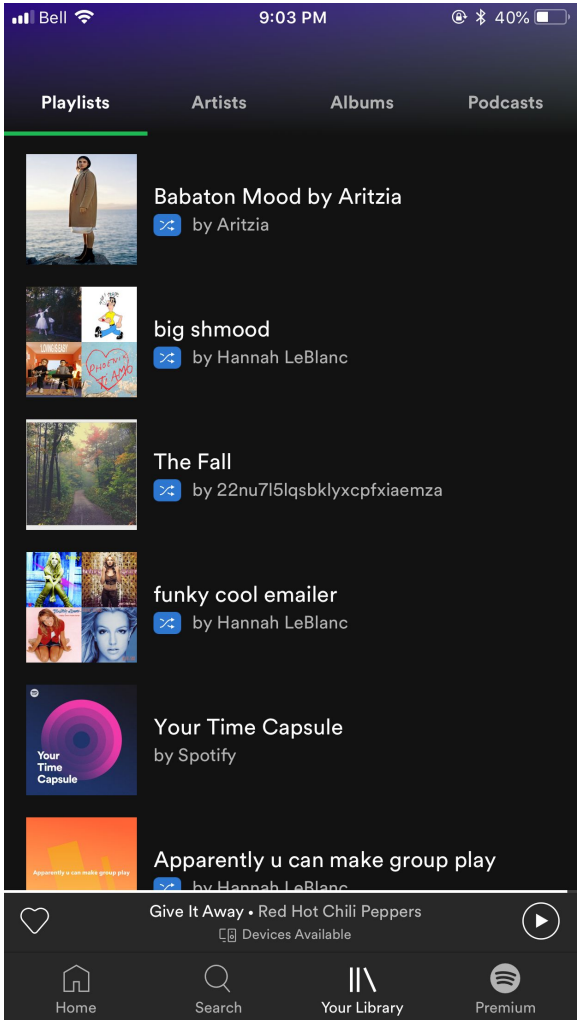
Let’s use OOP to describe a real-life concept ....

User Class

```
public class User {
      Public static String username = “hannahleblanc”;
      // there would be more attributes that describe me as an individual user

      // there could be different behaviours that are carried out through methods


      Public Friends[] getListofFriends() {

      }

      Private CC editCreditCardInfo() {

      }

      Public Playlists[] getPlaylists() {

      }
```

- My account can connect to other classes like a friends class that lets me add and delete friends.

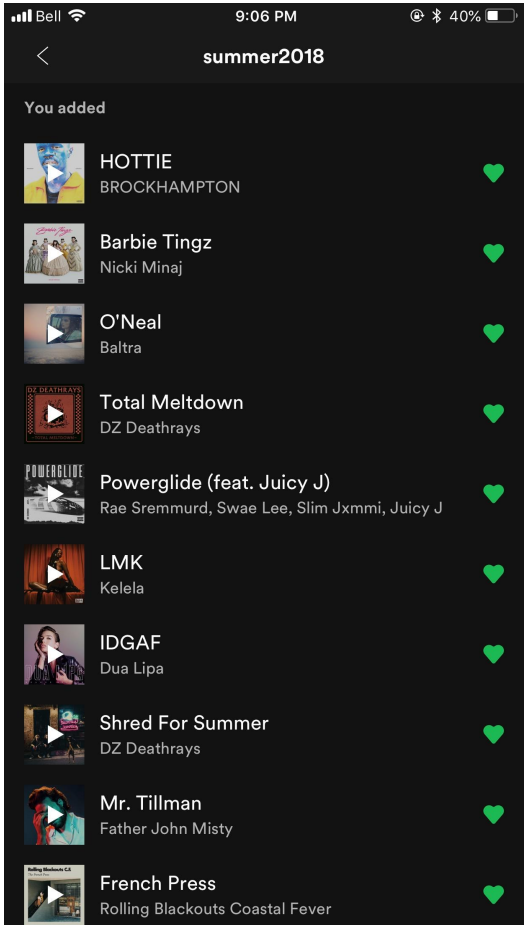
- Let’s look deeper at what a playlist class could look like and how we can control access to our playlists



A playlist class

- Possible attributes

- Possible actions/behaviours



Public Playlists

- Your playlist is open to the entire world and anyone can view it
- If you decided to collaborate with someone they can also edit the songs in your playlist.
- People can access your public playlist and use the public methods in the object class

Your account is private and you have friends.

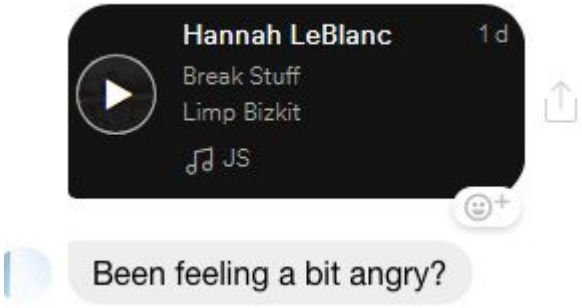
- Only people who you follow and trust to view your profile can view your playlists

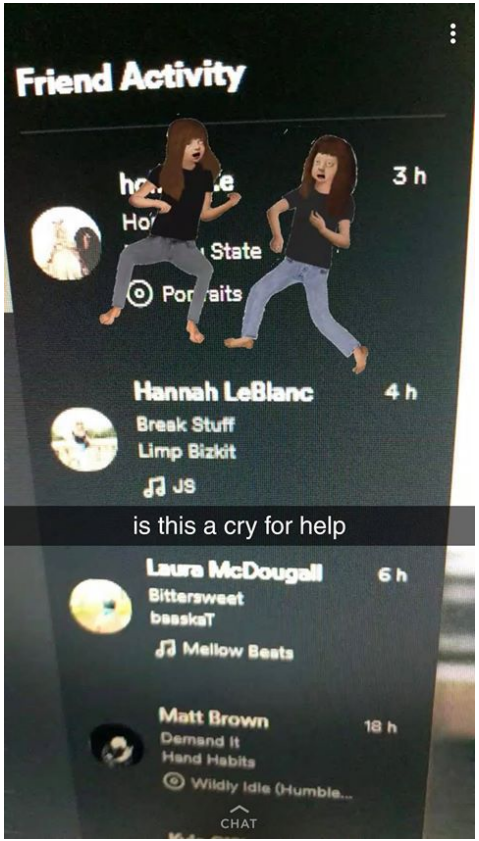
- The way that your circle of friends can see you is similar to the protected access modifier in Java where only other classes in the same package can see the class you’re working on.

- If your friends cannot edit your playlist then the methods in your class that allow you to add or delete songs would be set to private.

You listen to only your guilty pleasure songs and you don’t want anyone to know.

- Your account is private, you don’t add anyone as a friend.
- No one can view your private playlist objects and they definitely won’t be able to edit them.





when I study for 5 minutes straight
without checking my phone

Syntax Features

- All code in Java lives inside a class and is executed from inside functions
- { Curly braces show where a section of code begins and ends }
- Statements end with semi-colons ;
- All variables have declared types
- Functions MUST have a return type
- Void methods do not return anything

```
public static void main(String[] args)
```

Public - we can access this method

Static - this method isn’t associated with an instance or a particular object

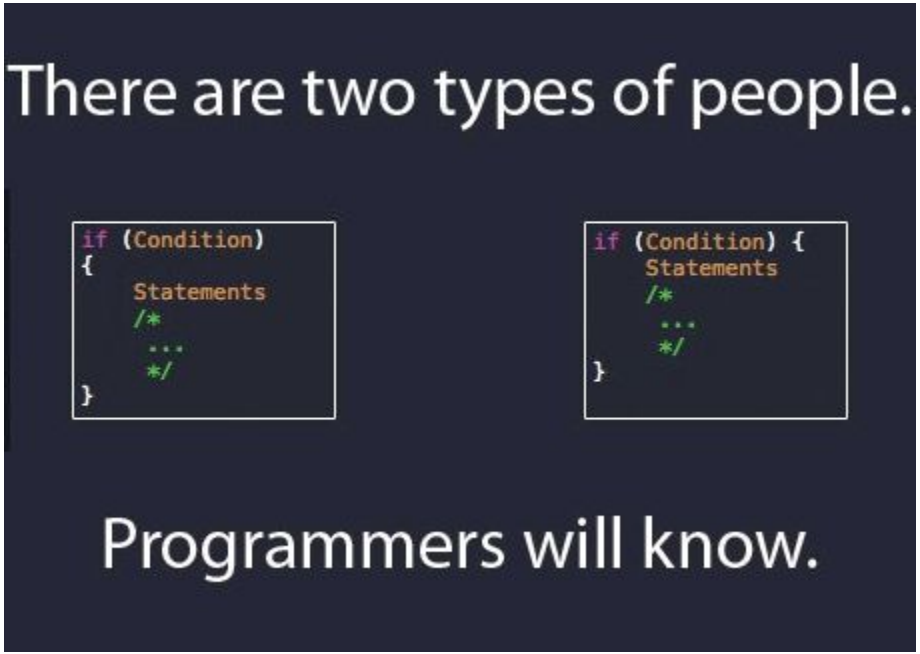
Void - main function doesn’t return anything

Main - we called this function main

Conditionals

```
int x = 5;

if (x <= 10) {
    x += 10;
    System.out.println(“What’s up I’m Hannah, I’m 23 and I never
learned how to code.);

} else { // x must be > 10
    x = x - 5;
}

if (x == 15)
    x -= 2;
System.out.println(x);
```

Curly Braces



CHAIN

```
if (condition) {

} else if (condition) {

} else if (condition) {

} else {

}
```

NESTED

```
if (condition) {

        If (condition) {

        } else {

        }

}
```

While

```
boolean imSpiraling = true;
int counter = 0;
while (imSpiraling) {
   if (counter >= 10)
      imSpiraling = false;
   counter += 1;
}
System.out.println(“~~~~I am calm now~~~~”)
```

For

```
public static int sum(int[] a) {
    int sum = 0;
    for (int i = 0; i < a.length; i = i + 1) {
        sum = sum + a[i];
    }
    return sum;
}

public static void main(String[] args) {
    int[] a = {4, 3, 2, 1};
    System.out.println(sum(a));
}
```

Doubles and strings

```
String greeting = ‘Hey what up hello’;

double num = 1738;

System.out.println(greeting);

System.out.println(double);

//OUTPUT

Hey what up hello

1738.0
```

Arrays

You

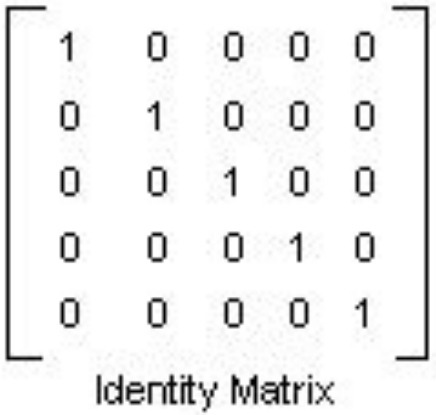
```
int[] numbers = new int[3];
numbers[0] = 4;
numbers[1] = 7;
numbers[2] = 10;
```

The guy she tells you not to worry about

```
int[] numbers = new int[]{4, 7, 10};
```

2D Arrays

```
int length = 5;

int[][] matrix = new int[length][length];

int row, col;

for(row=0; row<length; row++) {

      for(col=0; col<length; col++) {

            matrix[row][col] = row==col ? 1 : 0;

      }

}
```



Identity Matrix

2D Array

How do we change the values of just the top row?

```
int length = 5;

int[][] matrix = new int[length][length];

int row, col;

for(row=0; row<length; row++) {

      for(col=0; col<length; col++) {

            matrix[row][col] = row==1 ? 1 : 0;

      }

}
```

Good Java Resources

Derek Banas - Learn Java in 30 mins
https://www.youtube.com/watch?v=WPvGqX-TXP0&t=1096s

Code Academy - Java

https://www.codecademy.com/learn/learn-java

Oracle- Java Docs Tutorials

https://docs.oracle.com/javase/tutorial/java/index.html

1.

Write a Java function that takes two integers as input and divides them, return the result as a double.

2.

Replace all ‘f’ characters with ‘d’ characters in a given string. Print the length of the string. Return the string in all lowercase.

3.

Write a Java program to reverse an array of integer values.

4.

Rotate a 2D array by 90 degrees counterclockwise without using any extra storage.