Attendance

Start with refreshing on objects

1. Encapsulation - keep everything combined into one object
2. Example of car and its components - everything remains the same but one thing changes

```
class Phone {
  String model;
  String manufacturer;
  int memory;

  Phone(String model, String manufacturer, int memory) {
    this.model = model;
    this.manufacturer = manufacturer;
    this.memory = memory;
  }
}

void benifitObjects() {
  Phone S21_256 = new Phone("Galaxy S21", "Samsung", 256);
  Phone S21_128 = new Phone("Galaxy S21", "Samsung", 128);

  println(S21_128.state);
  S21_128.state = true;
  println(S21_128.state);
}
```

Problem with Arrays (don't tell the problem - walk through it)

1. Dynamic additions and removal
2. Don't tell the problem straight away let them discover it

```
void useArrays() {
  // Create an array of Phone objects
  Phone[] phones = new Phone[2];
  int phoneCount = 0;

  // Add phones to the array
  phones[phoneCount++] = new Phone("Galaxy S21", "Samsung",
256);
  phones[phoneCount++] = new Phone("Galaxy S21", "Samsung",
128);

  // Display phones using arrays
  for (int i = 0; i < phoneCount; i++) {
    Phone phone = phones[i];
    println("Phone Model: " + phone.model);
    println("Manufacturer: " + phone.manufacturer);
    println("Operating System: " + phone.os);
  }
}
```

Working with ArrayLists

1. Initalization
2. No size required

```
void useArrayLists() {
  // Create an ArrayList of Phone objects
  ArrayList<Phone> phones = new ArrayList<>();

  // Add phones to the ArrayList
  phones.add(new Phone("Galaxy S21", "Samsung", 256));
  phones.add(new Phone("Galaxy S21", "Samsung", 128));

  // Display phones using ArrayLists
  for (Phone phone : phones) {
    println("Phone Model: " + phone.model);
    println("Manufacturer: " + phone.manufacturer);
    println("Operating System: " + phone.os);
  }
}
```

Working with Primitives (don't tell the problem - walk through it)

1. Wrapper classes

Differences and benefits

```
void syntax() {
  String[] stringArray = new String[3];
  ArrayList<String> stringArrayList = new
ArrayList<String>();

  // adding elements
  stringArray[0] = "Apple";
  stringArray[1] = "Banana";
  stringArray[2] = "Cherry";

  stringArrayList.add("Apple");
  stringArrayList.add("Banana");
  stringArrayList.add("Cherry");


  // indexing elements
  println(stringArray[2]);
  println(stringArrayList.get(2));

  // getting length
  println(stringArray.length);
  println(stringArrayList.size());

  // setting elements
  stringArray[1] = "Blueberry";
  stringArrayList.set(1, "Blueberry");

  // possible to remove the last element but impossible
otherwise
  stringArray[1] = null;
  stringArrayList.remove(1);

  // printing
  // arrays need a for loop, this will work with processing
but with java will print by reference
  println(stringArray);
  println("ArrayList Contents: " + stringArrayList);

  // adding elements in the middle
  stringArrayList.add(1, "Blueberry");
  println("ArrayList Contents: " + stringArrayList);
}
```