Tim Tanasse
Java 210
April 6, 2015
Lab # 1: review of Java

- Define "class" and "Object"

    - Class:
      A collection of attributes and behaviors that describe something in real life.

    - Object:
      An instance of a class which has attributes and most if not all of the behavoirs of the parent class.

- What Java code is required for a class to properly implement the Comparable interface?

```java
public class MyClass implements Comparable<MyClass>
```

- What is the output of the code below

```java
double a = 0;
while (a <= 10){
  System.out.print(a + " ");
  a--;
}
```

Output does not end, prints decending 'a' variable by incriments of 1.0.

- Write a method that is called createArray that is passed a single integer value (guarenteed to be greater than zero). The method should create an array of type int, fill it with values starting at one, then return the array.

```java
public static int[] createArray(int size){
  int[] temp = new int[size];
  for(int i = 0; i < temp.length; i++){
    temp[i] = i + 1;
  }
  return temp;
}
```

- Write the six standard methods every self-respecting class should have for the Song class

```java
public String getName(){ return name; }

public void setName(String name){ this.name = name; }

public String getArtist(){ return artist; }

public void setArtist(String artist){ this.artist = artist; }

@Override
public boolean equals(Object obj){
  if(this == obj)
    return true;
  if (obj == null)
    return false;
  if (this.getClass() != obj.getClass())
    return false;
  Song that = (Song)obj;
  if (this.artist != that.artist)
    return false;
  if (this.name != that.name)
    return false
  return true;
}
@Override
public String toString(){
  return artist + " - " + name;
}

public Song(){
  this("Untitled", "Undetermined")
}
public Song(String name, String artist){
  this.name = name;
  this.artist = artist;
}
```

- Which is more efficient for an array of elements:

  - insertionSort or selectionSort? Why?
    insertionSort has the same time complexity as selectionSort for a completely unsorted array, but the time complexity improves as arrays are more partially or completely sorted, where the time complexity does not improve for partially sorted arrays for selectionSort, and only improves for fulling sorted arrays.

  - linearSearch or binarySearch? Why?
    binarySearch, in general. The time complexity is far better, but due to the cost of it needing to be sorted, linearSearch may be better if the array doesn't need to be sorted, or if the array isn't getting searched more than a few times.

- What condition must be met for binarySearch?
  The array must be sorted for binarySearch to work.

- What is the time complexity formula for binarySearch?

$$(O) \log_2 n$$