# CSE 143: Section AA

TA: Eric Lemar

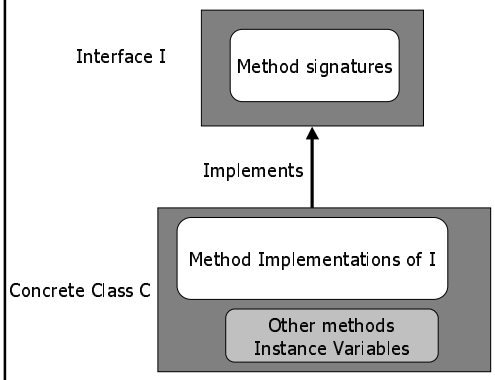
## Interfaces

- Interfaces describe class method signatures
  - ie, it lists methods a class MUST have
- An interface is NOT a class itself
  - It provides no implementation
  - An object can't be made from it

```
interface SquareInterface
{
  public int getSideLength();
   public void setSideLength(int length);
}
```

- *Using an interface:*

```
class MySquare implements SquareInterface {
    private int sideLength;
    public MySquare() {
      sideLength = 1;
    }
    public int  getSideLength() {
      return sideLength;
    }
    public void setSideLength(int length) {
      sideLength = length;
    }
   //Note: we can have EXTRA functions
    public int  getArea() {
      return sideLength*sideLength;
    }
}
```




## Interface

- We can use an interface many times

```
class MySquare implements SquareIterface {
...
}

class YourSquare implements SquareInterface {
....
}
```

## Why use interfaces?

- Specify classes before they are implemented
  - The compiler makes sure we have all necessary functions
  - The compiler CANNOT see that the functions do the right things
- Other software engineering reasons
  - We'll get into these more latter

## Exercise

*Write a method to print an array of strings:*

```
void printStringArray(String names[]) {
...
}
```

## Exercise

```
void printStringArray(String names[]) {
   if (null != names) {
      int i;
      for (i=0;i<names.length;i++) {
        System.out.println(names[i]);
      }
   }
}
```

## ArrayList

```
import java.util.*;

void printStringArrayList(ArrayList names)
{
    if (null != names) {
       Iterator it=names.iterator();
       while (it.hasNext()) {
          String str = (String)it.next();
          System.out.println(str);
       }
    }
}
```

## Arraylists

```
arr = new ArrayList();
....
arr.get(i)
arr.set(i,newValue)
arr.add(newValue)
```