CSCI 132:
Basic Data Structures and Algorithms

Intro to Java (Loops, Arrays)

Reese Pearsall & Iliana Castillon

Fall 2024

Announcements

- Lab 1 due **tomorrow** at 11:59 PM

→ After today, you will be able to complete it

→ Submit .java files (don't rename them)

I'll be posting this week's materials in a Brightspace announcement

Student Success Center - Fall 2024

Tutoring Schedule - Barnard Hall 259

Fall Semester tutoring begins on Monday, August 26th. Barnard 254/259 is generally available 24/7.

Schedule	Monday	Tuesday	Wednesday	Thursday	Friday
8:00 a.m.					
9:00 a.m.				Oscar Oropeza	
10:00 a.m.	Gabriel Martens				Caleb Eardley Anthony Nania
11:00 a.m.	Sundas Iftikhar Dominick Valenti	Fatima Ododo Angelo Porcella	Ismoiljon Muzaffarov Andras Necz	Angelo Porcella Gideon Popoola	Caleb Eardley
Noon	Sundas Iftikhar	Fatima Ododo Gerard Shu Fuhnwi	Ismoiljon Muzaffarov Andras Necz	Sultan Yarylgassimov Joseph Windmann	Riley Slater
1:10 p.m.	Matt Ivankovich AJ Zetzer	Michael Belmear	Turner Burchard	Sultan Yarylgassimov	Riley Slater Gerard Shu Fuhnwi
2:10 p.m.	Dillon Shaffer	Will Mitchell	Turner Burchard Ben Logan	Sean Newsome	
3:10 p.m.	Justin Mau Gideon Popoola	Nishu Nath	Alex Ellingsen	Felicia Jayasaputra	Dillon Shaffer
4:10 p.m.	Justin Mau	Nishu Nath Karishma Rahman	Jack Hayward Karishma Rahman	Felicia Jayasaputra	
5:10 p.m.					

Example: A student is allowed to register for CSCI 476 if they have a GPA greater than 2.0, **and** if they are a Junior **or** Senior

```
public void allowToRegister() {

    if (this.gpa > 2.0) { // check the first condition (Alternatively, we could use an && here)

        if (this.year.equals("Junior") || this.year.equals("Senior")){

            System.out.println("Student is allowed to register for CSCI 476");

        }
    }
}
```

Student.Java

We can check one of two conditions is true using the or operator (||)

(we do not have the `or` keyword in Java)

```
student1.determineYear();
```

StudentDemo.Java

Example: A student is allowed to register for CSCI 476 if they have a GPA greater than 2.0, **and** if they are a Junior **or** Senior

```
public void allowToRegister() {

    if (this.gpa > 2.0) { // check the first condition (Alternatively, we could use an && here)

        if (this.year.equals("Junior") || this.year.equals("Senior")){

            System.out.println("Student is allowed to register for CSCI 476");

        }
    }
}
```

Student.Java

Why do `this.year.equals("Junior")` and not `this.year == "Junior"`

Checking for string equality in Java is a little bit funky…

Using == does **not** check for equivalence of values between two strings…

Example: A student is allowed to register for CSCI 476 if they have a GPA greater than 2.0, **and** if they are a Junior **or** Senior

```
public void allowToRegister() {

    if (this.gpa > 2.0) { // check the first condition (Alternatively, we could use an && here)

        if (this.year.equals("Junior") || this.year.equals("Senior")){

            System.out.println("Student is allowed to register for CSCI 476");

        }
    }
}
```

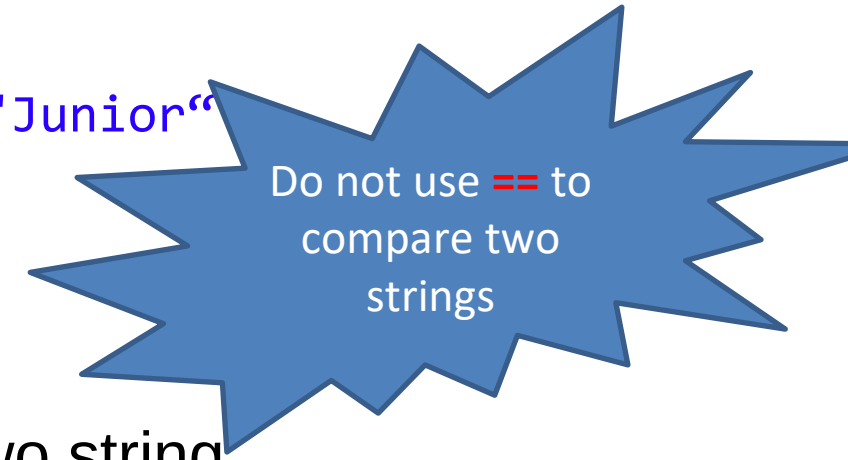
Student.Java

Why do `this.year.equals("Junior")` and not `this.year == "Junior"`?

Checking for string equality in Java is a little bit funky…

Using == does **not** check for equivalence of values between two strings…

Instead, we need to use the **.equals()** method between two string



Arrays are a *collection* of data

→ Once initialized, are **fixed** in size

→ Can only hold one data type

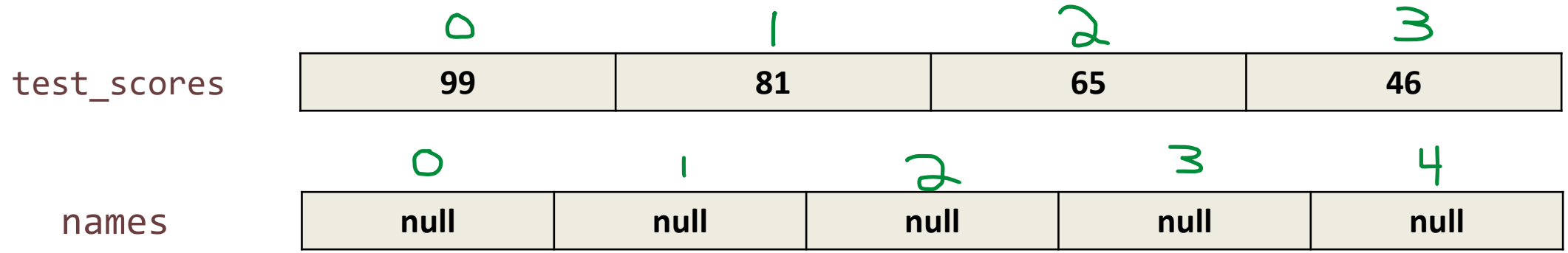
```
System.out.println(test_scores[2]);
>> 65

System.out.println(test_scores[4]);
>> ERROR
```

Declaring an array and giving it a value

```
int[] test_scores = {99, 81, 65, 46};
```

Declaring an array allocating 5 empty spots (we need to fill them later)

```
String[] names = new String[5];
```



For loops can be used to iterate across an array.

Two ways:

1. Iterate by index

```
String[] animals = {"Zebra", "Elephant", "Lion", "Penguin"};

for (int i = 0; i < animals.length; i++) {

  System.out.println(animals[i]);

}
```

2. Iterate by element

For loops can be used to iterate across an array.

Two ways:

1. Iterate by index

```
String[] animals = {"Zebra", "Elephant", "Lion", "Penguin"};

for (int i = 0; i < animals.length; i++) {
  System.out.println(animals[i]);

}
```

Start at index 0

stop at index 4 (length of array)

Increase the index by 1 each time

Each for loop has:

1. A start
2. A stop
3. A step

2. Iterate by element

For loops can be used to iterate across an array.

Two ways:

1. Iterate by index

```
String[] animals = {"Zebra", "Elephant", "Lion", "Penguin"};

for (int i = 0; i < animals.length; i++) {

  System.out.println(animals[i]);

}
```

2. Iterate by element

```
for (String i : animals) {
    System.out.println(i);
}
```

Both will give you the exact same output…