

More Loops



Department of Computer Science

Slides Originally Created by Albert Lionelle
(Albert.Lionelle@colostate.edu), updated by Marcia Moraes
(marcia.moraes@colostate.edu)

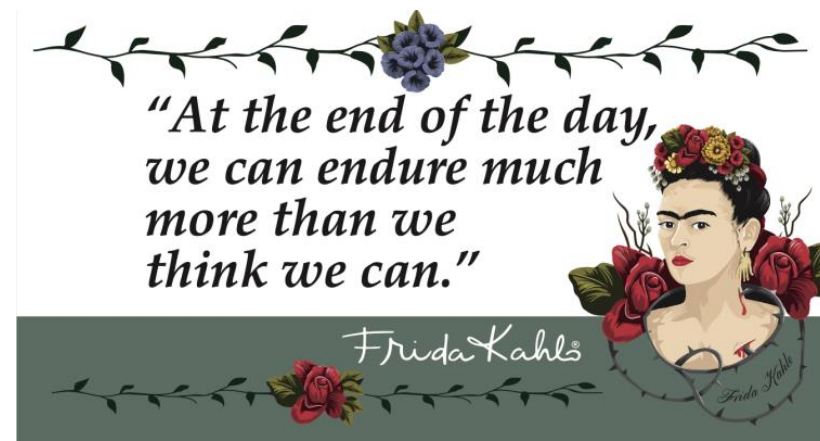
Announcements

TODO Reminders:

Readings are due **before** lecture

- Reading 14 (zybooks) – you should have already done that 😊
- Lab 09
- Reading 15 (zyBooks) – you should have already done that 😊
- Lab 10
- Reading 16 (zybooks)
- RPA 7

Keep practicing your RPAs in a spaced and mixed manner 😊

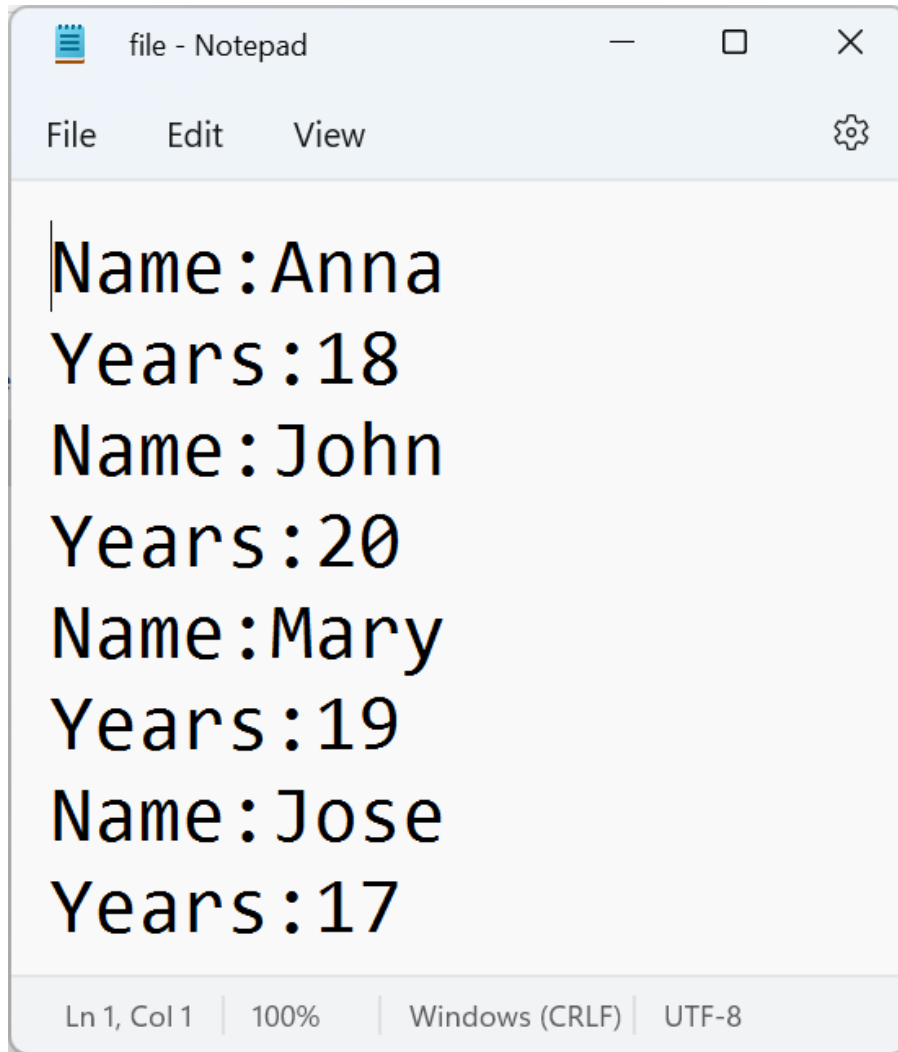


<https://www.linkedin.com/pulse/end-day-we-can-endure-much-more-than-think-beatriz-alvarado/>

NEXT WEEK Exam 2

Don't procrastinate
and catch up if you
need!

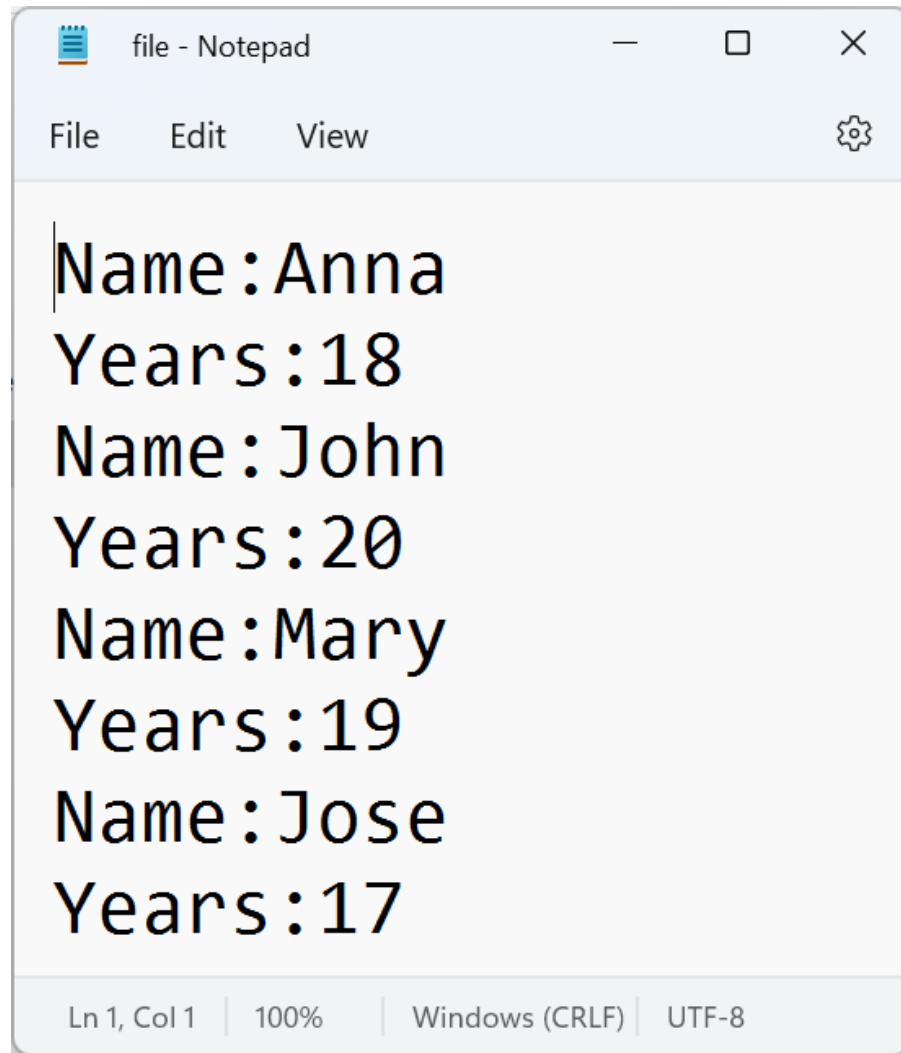
Review – How to read and print from a file

A screenshot of a Notepad window titled "file - Notepad". The window has a menu bar with "File", "Edit", and "View". The text content is as follows:
Name:Anna
Years:18
Name:John
Years:20
Name:Mary
Years:19
Name:Jose
Years:17
At the bottom, the status bar shows "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

```
file - Notepad
File Edit View
Name:Anna
Years:18
Name:John
Years:20
Name:Mary
Years:19
Name:Jose
Years:17
Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

```
public static void readFile(String fileName){
    try {
        Scanner fileIn = new Scanner(new File(fileName));
        while(fileIn.hasNext()) {
            String line = fileIn.nextLine();
            System.out.println(line);
        }
    }catch(IOException ex) {
        System.err.print("Error reading file!");
        ex.printStackTrace();
    }
}
```

Review – How to parse the String to get specific content

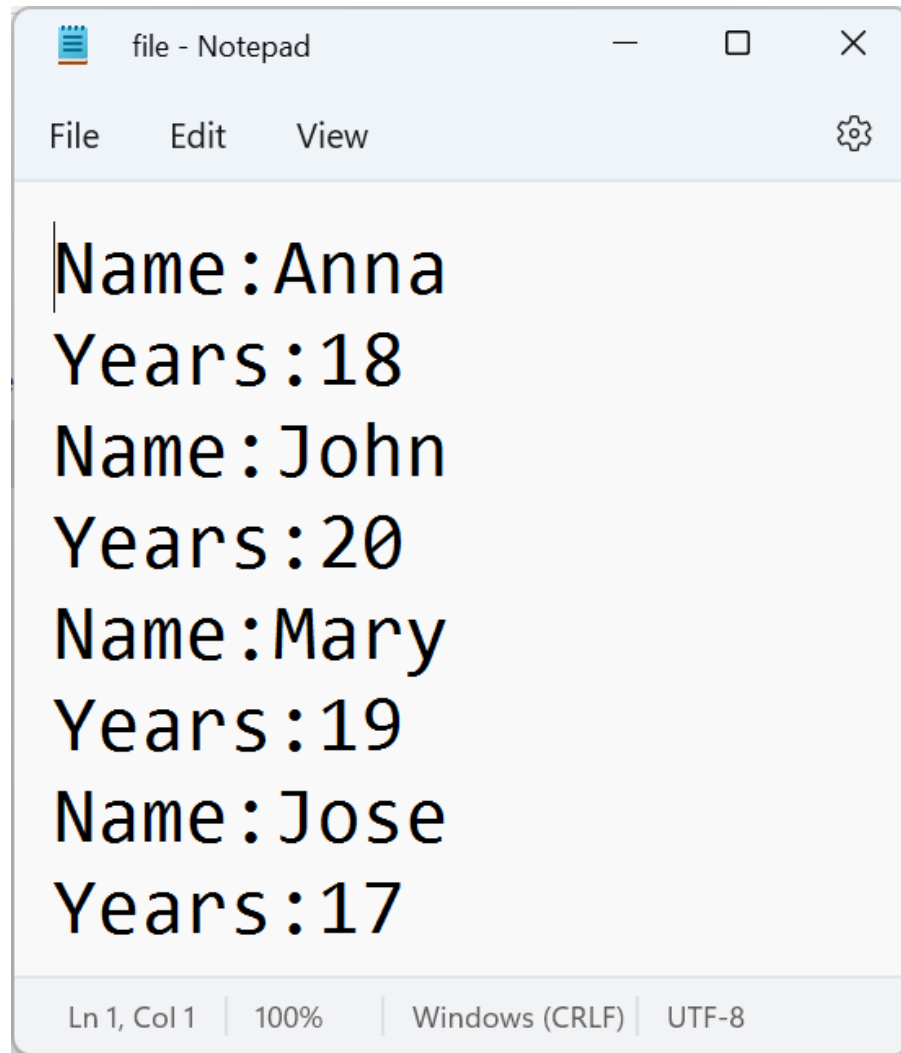


```
file - Notepad
File Edit View
Name:Anna
Years:18
Name:John
Years:20
Name:Mary
Years:19
Name:Jose
Years:17
Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

```
public static void readFile(String fileName){
    try {
        Scanner fileIn = new Scanner(new File(fileName));
        while(fileIn.hasNext()) {
            String lineName = fileIn.nextLine();
            String lineYear = fileIn.nextLine();
            String name = parse(lineName);
            String year = parse(lineYear);
            System.out.println(name);
            System.out.println(year);
        }
    } catch(IOException ex) {
        System.err.print("Error reading file!");
        ex.printStackTrace();
    }
}

public static String parse(String line) {
    //.trim() removes any spaces that we have at the
    //begin or end of the string
    return line.substring(line.indexOf(":") + 1).trim();
}
```

Review – How to create an object from the specific content



A screenshot of a Notepad window titled "file - Notepad". The window contains the following text:

```
Name:Anna
Years:18
Name:John
Years:20
Name:Mary
Years:19
Name:Jose
Years:17
```

The status bar at the bottom indicates "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

```
public static ArrayList<Person> readFile(String fileName){
    ArrayList<Person> lst = new ArrayList<>();
    try {
        Scanner fileIn = new Scanner(new File(fileName));
        while(fileIn.hasNext()) {
            String lineName = fileIn.nextLine();
            String lineYear = fileIn.nextLine();
            String name = parse(lineName);
            String year = parse(lineYear);
            Person p = new Person(name, Integer.parseInt(year));
            lst.add(p);
        }
    } catch(IOException ex) {
        System.err.print("Error reading file!"); ex.printStackTrace();
    }
    return lst;
}

public static String parse(String line) {
    //.trim() removes any spaces that we have at the
    //begin or end of the string
    return line.substring(line.indexOf(":") + 1).trim();
}
```

Recall Activity – Different Types of Loops

- What is the output of each one of those loops?

1)

```
for(int x = 0; x < 100; x++) {  
    System.out.println(x);  
}
```

2)

```
int count = 10;  
while(--count > 0) {  
    // block of code to repeat  
    // notice, --count is valid  
    System.out.print(count + " ");  
}
```

3)

```
int counter = 10;  
String tracker = "Track:";  
do {  
    tracker += " " + counter;  
}while(++counter < 11);  
System.out.println(counter);  
System.out.println(tracker);
```

What is printed in each case?

Why that happened?

```
1) int counter = 10;  
   String tracker = "Track:";  
   do {  
       tracker += " " + counter;  
   }while(++counter < 11);  
   System.out.println(counter);  
   System.out.println(tracker);
```

11

Track: 10

```
2) int counter = 10;  
   String tracker = "Track:";  
   do {  
       tracker += " " + counter;  
   }while(counter++ < 11);  
   System.out.println(counter);  
   System.out.println(tracker);
```

12

Track: 10 11

Incrementor and Decrementor

- ++ or -- BEFORE the variable means
 - add or subtract by 1, and then use the modified variable
- ++ or -- AFTER the variable means
 - use the value in the variable, and then add or subtract by one modifying the variable

Incrementor Example – What is printed?

```
public class TrackerExample {  
    private ArrayList<String> values;  
    private int current = 0;  
    public TrackerExample(ArrayList<String> toTrack) {  
        values = toTrack;  
    }  
    public String getCurrent() {  
        return values.get(current++);  
    }  
}
```

```
Scoobert "Scooby" Doo  
Norville "Shaggy" Rogers
```

```
public class AppTracker {  
    public static void main(String args[]){  
        ArrayList<String> students = new ArrayList<>();  
        students.add("Scoobert \"Scooby\" Doo");  
        students.add("Norville \"Shaggy\" Rogers");  
        students.add("Daphne Blake");  
        students.add("Velma Dinkley");  
        students.add("Fred Jones");  
        TrackerExample tracked =  
            new TrackerExample(students);  
        System.out.println(tracked.getCurrent());  
        System.out.println(tracked.getCurrent());  
    }  
}
```

Are those methods equivalent?

```
public String getCurrent() {  
    return values.get(current++);  
}
```

```
public String getCurrent() {  
    String tmp = values.get(current);  
    current += 1;  
    return tmp;  
}
```

Do While Loop

- A unique loop
 - For and While – both check and then run
- Do-While
 - runs block of code and **then** checks
 - Guarantees at least one run

Format:

```
do {  
    //code block  
} while(condition);
```

loops while condition is true

```
Scanner scnr = new Scanner(System.in);  
int x;  
do {  
    System.out.print("Please enter a number: ");  
    x = scnr.nextInt();  
    System.out.println();  
}while(x < 10);
```

What this do...while is doing?

It is validating the number read, the number needs to be greater or equal than 10 to be valid.

Changing loop order: Break / Continue

- `break`
 - allows to break out of the loop completely (early exit)
- `continue`
 - allows us to move back to the top of the code block
 - with for loop, the incrementor is still completed

```
for(int x = 0; x < 100; x++) {  
    if(x == 10) break;  
    System.out.print(x);  
}
```

0123456789

```
for(int x = 1; x < 10; x++) {  
    if(x%2 == 0) continue;  
    System.out.print(x);  
}
```

13579

Nesting Loops

- What is printed?

```
for (char c = 'A'; c < 'D'; c++) {  
    System.out.print("Seat:");  
    for (int i = 0; i < 2; i++) {  
        System.out.print(" " + c + i);  
    }  
    System.out.println();  
}
```

Seat: A0 A1

Seat: B0 B1

Seat: C0 C1

What code would I want to change if I wanted to make 3 seats per row?

What code would I change if I wanted to add another row?

Do the Worksheet for Today's class

- Worksheet is also available on Canvas