Announcements

● Assignment 1 was due at **10am this morning**.
● Assignment 2 is going out right after lecture!
  ○ Assignment 2 is due at **10am on Wednesday, July 10th**.
● No lecture or sections Thursday for 4th of July.
  ○ If your section in cancelled, please try to attend a Wednesday section or Friday’s section (11:30am in Skilling Auditorium)
● No LaIR Wednesday, July 3rd due to Holiday.
Announcements

- When emailing instructors, *always email both of us!*
  - We have a joint email: cs106a-sum1819-staff@lists.stanford.edu

- If you don’t have a discussion section, please email both instructors ASAP to be placed in a section.
Plan for Today

- Review: Methods
- Nested For Loop with Ascii
- Intro to Graphics Program - Our First Graphics Program!
- Checkerboard
- Infinite Loops
/* We will make Java a sandwich.
 * Pre: n/a
 * Post: We will have a completed sandwich.
 */

private food makeASandwich(food bread, food veg, food protein, food spread){
    food sandwich = bread + veg + protein + spread + bread;

    return sandwich;
}
/* We will do x, y, and z actions and return methodType data
* Pre: What we assume is true beforehand.
* Post: What we promise is true afterwards.
*/

visibility methodType methodName(paramType1 paramName1, paramType2 paramName2){
    commands;

    return dataThatIsMethodType;
}
public void run(){
    double average = calculateAverage(5, 10);
    println("The average is: "+average);
}

private double calculateAverage(double num1, double num2){
    println("I can also do actions!");
    println("I also HAVE to return a double because that’s the method type!");

    double sum = num1 + num2;
    return sum / 2; // This will equal 7.5
}
public void run()
{
    double average = 7.5;
    println(“The average is: ” + average);
}

private double calculateAverage(double num1, double num2)
{
    println(“I can also do actions!”);
    println(“I also HAVE to return a double because that’s the method type!”);
    
    double sum = num1 + num2;
    return sum / 2; // This will equal 7.5
}
Scope is the idea that variables only exist inside a certain block of code.

In Java, a variable is born when it is declared.

A variable terminates when it hits the ending bracket of the code block in which it was declared.

```java
public void run(){
    int newVariable = 0; // I am born!
    ...
    ...
    ...
} // I hope I haven’t bored you. Goodbye.
```
The variable only exists from its declaration to the end of its current code block.

```java
while (conditionAIsTrue){
    if (conditionBIsTrue){
        ...
        int newVariable = 0; // I am born!
        ...
    }
    // I’m bored with it all. Goodbye.
}
```
A Variable Love Story: Scope

It doesn’t exist before it’s declared, and it doesn’t exist outside of its current code block!

```java
while (conditionAIsTrue){
    if (conditionBIsTrue){
        ...
        int newVariable = 0; // I am born!
        ...
    }
    // I’m bored with it all. Goodbye.
}
println(newVariable);
```
Review: For Loop

This code is run once, just before the for loop starts

This code is run each time the code gets to the end of the “body”

Repeats the loop if this condition passes

```
for (int i = 0; i < 3; i++) {
    println("I love CS106A!");
}
```

I love CS106A!
I love CS106A!
I love CS106A!
Dot Art

How can we do this?
How can we do this?

We use for loops to repeat an action a set number of times.

Let’s try it!
**Pseudocode:**
Repeat 10 times:

Draw a line of 7 stars

```java
public void run(){
    drawDotArt();
}

private void drawDotArt(){
}
```
Pseudocode:
Repeat 10 times:
  Draw a line of 7 stars

```java
public void run(){
    drawDotArt();
}
private void drawDotArt(){
    for (int i = 0; i < 10; i++) {
    }
}
```
Pseudocode:
Repeat 10 times:
   Draw a line of 7 stars

```java
public void run(){
    drawDotArt();
}

private void drawDotArt(){
    for (int i = 0; i < 10; i++) {
        println("* * * * * *");
    }
}
```
Pseudocode:
Repeat 10 times:

**Draw a line of 7 stars**

```
public void run(){
    drawDotArt();
}

private void drawDotArt(){
    for (int i = 0; i < 10; i++) {
        println("* * * * * * *");
    }
}
```

This line seems a little repetitive. How can we use code to make this less repetitive?
**Pseudocode:**
Repeat 10 times:
  Repeat 7 times:
    Draw a star
    Make a new line

```java
public void run(){
    drawDotArt();
}

private void drawDotArt(){
    for (int i = 0; i < 11; i++) {
        println("* * * * * *");
    }
}
```
Dot Art: Nested For Loops

Pseudocode:
Repeat 10 times:
  Repeat 7 times:
    Draw a star
    Make a new line

```java
public void run(){
    drawDotArt();
}

private void drawDotArt(){
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 7; j++) {

        }
    }
}
```
Pseudocode:
Repeat 10 times:
  Repeat 7 times:
    Draw a star
    Make a new line

```java
public void run(){
    drawDotArt();
}

private void drawDotArt(){
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 7; j++) {
            print("*");
        }
    }
}
```
Dot Art: Nested For Loops

**Pseudocode:**
Repeat 10 times:
  Repeat 7 times:
    Draw a star
    Make a new line

```java
public void run(){
    drawDotArt();
}

private void drawDotArt(){
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 7; j++) {
            print("*");
        }
        println();
    }
}
```
We’ve added parameters so we can take our user input and pass it into our method. 

r will equal rows and c will equal cols.
Can Dots be Art?
Isn’t This Art???

Not really the same, but okay...
Speaking of Art...
Let’s Look at Some Graphics Programs!

Program

- Karel Program
- Console Program
- Graphics Program
Let’s Look at Some Graphics Programs!
As We’re Learning: JavaDocs!

CS 106A: Programming Methodology
Summer 2019
Monday, Tuesday, Wednesday, Thursday 10:30AM-11:20AM PST in Bishop Au

RESOURCES
- Course Schedule
- Style Guide
- Piazza
- Eclipse
- Paperless
- Textbooks
- Stanford Java Docs
- Pair Programming
- Course Staff

NEW ANNOUNCEMENTS

July 4 Holiday
2 days ago
As a reminder, because of the July 4 holiday, we will have no class.
Additionally, Thursday sections are cancelled. If you are unable to attend either a Wednesday section (times/locations) OR the Friday (July 5) section at 11:30 am this week due to the Holiday.

Overview
Package Class Tree Index Help

PREV NEXT FRAMES NO FRAMES

The ACM Java Libraries

Package acm.graphics
This package provides a set of classes that support the creation of simple, object-oriented graphical displays.

Package acm.gu
This package provides a set of classes that support the creation of simple, interactive programs.

Package acm.graphics
This package provides a set of classes that support the creation of simple, object-oriented graphical displays.

Overview
Package Class Tree Index Help

PREV NEXT FRAMES NO FRAMES

The ACM Java Libraries

Package acm.graphics
This package provides a set of classes that support the creation of simple, object-oriented graphical displays.

Package acm.gu
This package provides a set of classes that support the creation of simple, interactive programs.

Interface Summary

GContainer Defines the functionality of an object that can serve as the parent of a GObject.
GFillable Specifies the characteristics of a graphical object that supports filling.
GResizable Specifies the characteristics of a graphical object that supports the setSize and setBounds methods.
G Scalable Specifies the characteristics of a graphical object that supports the scale method.

Class Summary

G3DRect The G3DRect class is used to represent a rectangle whose borders are drawn to create a three-dimensional effect.
GArc The GArc class is a graphical object whose appearance consists of an arc.
GCanvas The GCanvas class is a lightweight component that also serves as a container for graphical objects.
GCompound This class defines a graphical object that consists of a collection of other graphical objects.
GDimension This class is a double-precision version of the Dimension class in java.awt.
GImage The GImage class is a graphical object whose appearance is defined by an image.
We don’t expect you to memorize everything! Use the graphics library documentation to look up methods you can use in graphics programs!
A GRect is a variable type that stores a rectangle.

As an example, the following code displays a blue square.

```java
public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
**Our First Graphics Object: GRect**

```java
public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
Our First Graphics Object: GRect

```java
public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```
Our First Graphics Object: GRect

public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
Our First Graphics Object: GRect

```java
public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```

rect
Our First Graphics Object: GRect

```java
public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
```

rect
Our First Graphics Object: GRect

public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}

Note: The GRect is created, but hasn’t been added to the screen.
public void run(){
    drawBlueSquare();
}

private void drawBlueSquare(){
    GRect rect = new GRect(200, 200);
    rect.setFilled(true);
    rect.setColor(Color.BLUE);
    add(rect, 50, 50);
}
Note: The y coordinate gets bigger as we go down. This is opposite of how it acts in most math classes!
Our First Graphics Challenge
Checkerboard
Our First Graphics Challenge

What’s the Pseudocode?
What’s the Pseudocode for drawing the first row of boxes?
What’s the Pseudocode for drawing the first row of boxes?

repeat for 8 columns:
  draw a box
  (x location is col# * box size)
  (y location is 0)
  add the box to the screen
How can we change the Pseudocode to draw all of the boxes?

repeat for 8 columns:
  draw a box
  (x location is col# * box size)
  (y location is 0)
  add the box to the screen
Our First Graphics Challenge

Pseudocode for drawing all of the boxes:

repeat for 8 rows:
  repeat for 8 columns:
    draw a box
    (x location is col# * box size)
    (y location is row# * box size)
    add the box to the screen
Our First Graphics Challenge

How can we change the Pseudocode to fill in the correct boxes?

repeat for 8 rows:
  repeat for 8 columns:
    draw a box
    (x location is col# * box size)
    (y location is row# * box size)
    add the box to the screen
Our First Graphics Challenge

Pseudocode for filling the correct boxes:

repeat for 8 rows:
    repeat for 8 columns:
        draw a box
        \((x \text{ location is } \text{col#} \times \text{ box size})\)
        \((y \text{ location is } \text{row#} \times \text{ box size})\)
        If row+column is odd:
            color the box black
            add the box to the screen
Let’s Code It!
What’s an infinite loop?
Infinite loops are loops that never end.

Why would this ever happen?
What is wrong with this loop?

```java
int answer = readInt("What is 1 + 1?");
while (answer != 2){
    println("Wrong answer! Try again");
}
```
Example: Infinite Loops

What is wrong with this loop?

It doesn’t give the user a chance to change their answer.

```java
int answer = readInt("What is 1 + 1?");
while (answer != 2){
    println("Wrong answer! Try again");
}
```
What is wrong with this loop?

It doesn't give the user a chance to change their answer.

This means the while loop will never end! This is effectively a while(true) loop!

```java
int answer = readInt(“What is 1 + 1?”);
while (answer != 2){
    println(“Wrong answer! Try again”);
}
```
Example: Infinite Loops

How can we fix it?

```java
int answer = readInt("What is 1 + 1?");
while (answer != 2){
    println("Wrong answer! Try again");
}
```
Example: Infinite Loops

How can we fix it?

We can add a line where we give the user a chance to change their answer!

```java
int answer = readInt("What is 1 + 1?");
while (answer != 2){
    println("Wrong answer! Try again");
    answer = readInt("What is 1 + 1?");
}
```
Plan for Today

- Review: Methods
- Nested For Loop with Ascii
- Intro to Graphics Program - Our First Graphics Program!
- Checkerboard
- Infinite Loops

Reminders:
- Assignment 2 is due at **10am on Wednesday, July 10th**.
- No lecture or sections Thursday for 4th of July.
- No LaIR Wednesday, July 3rd due to Holiday.