Instructions to Participate in Today’s Session

To send your answers to today’s questions, you can access Top Hat:

- As a guest

Or

- Registered User

1. If you want to participate as a guest

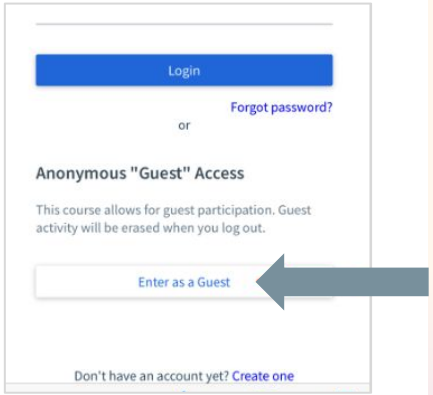
Mobile Instructions

01

Scan the **QR Code**

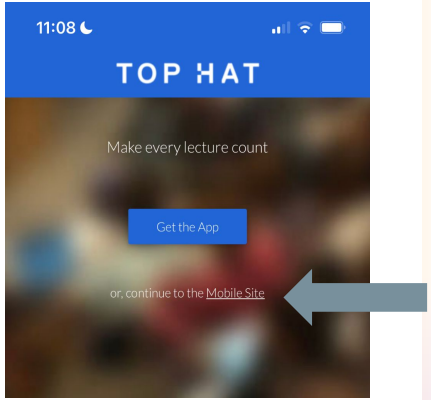
02

Scroll down and select **'enter as guest'**



03

Select **'continue to mobile site'**

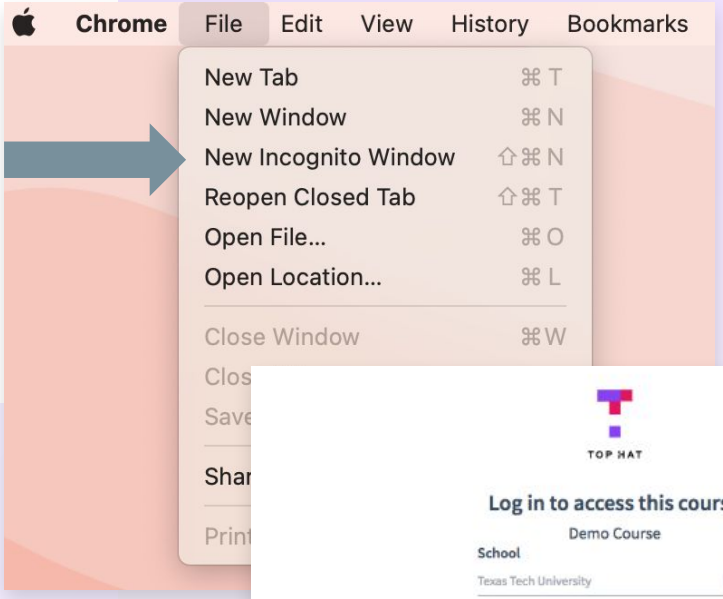


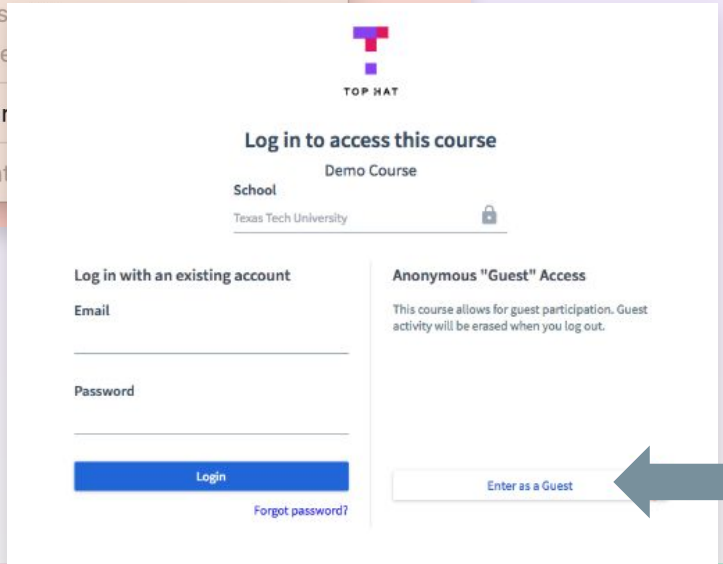
1. If you want to participate as a guest

Laptop/Tablet **Instructions**

Follow these instructions to access a temporary student account to participate today!

1. On your device, navigate to a Private or Incognito Browser Window
2. Enter https://app.tophat.com/e/170052
3. Select **'Enter as Guest'** on the right hand side of the screen
4. You're in! Participate as a student in today's session





2. If you want enter as a registered user

1. Navigate to app.tophat.com or download and open the app
2. Enter University of Arizona as the school
3. Sign in with your Net ID
4. Search for class with the join code **170052** and start participating

CS 101
Code Style

Appendix A

Question 3

1. Open your textbook and go to Appendix A
https://learning.oreilly.com/library/view/make-getting-started/9781457187070/app01.html

2. Skim Appendix A to answer the following question

3. How does processing works with respect to Spaces, Uppercase and Lowercase, and Style?

Organization

- Using good programming style is important
  - Especially as program size grows
- In general (not just in code) things are easier to use and understand when they are in-order
- Good style is crucial to maintaining a code base

ICA

Which has better style?

A

```
void draw() {
  int red = 50;
  for (int i = 20; i < 220 ; i += 60) {
    fill(red, 20, 20);
    red = red + 70;
    for(int j = 5 ; j < 500 ; j += 50) {
      rect(j, i + 20, 40, 40);
    }
  }
}
```

B

```
void  draw ( )
{        int red = 50;
  for (
    int i = 20; i < 220 ;
i += 60) { fill(red, 20, 20);
        red = red + 70;
 for(int j = 5 ; j < 500 ; j += 50)
{
      rect(   j, i + 20,    40,  40);
  }
  }
   }
```

C - They both look good to me.

Which has better style?

ICA

```
void draw( )
        { fill(255, 100, 25);
      rect(0, 0,        300, 100);
fill(255, 255, 0);
      rect(0,  100, 300,   100);
  fill(0,  255,   255);
 rect (0,  200, 300, 100);
if ( mousePressed) {
          if (mouseY < 100 )  {
  link("http://www.amazon.com")  ;
   }   else     if (mouseY <     200)
 {
         link("http://www.google.com");
   } else

{
     link("http://www.espn.com");
    }   }}
```

```
void draw() {
  fill(255, 100, 25);
  rect(0, 0, 300, 100);
  fill(255, 255, 0);
  rect(0, 100, 300, 100);
  fill(0, 255, 255);
  rect(0, 200, 300, 100);
  // handle mouse clicks to take user to URL
  if(mousePressed) {
    if (mouseY < 100) {
      link("http://www.amazon.com");
    } else if (mouseY < 200) {
      link("http://www.google.com");
    } else {
      link("http://www.espn.com");
    }
  }
}
```

Which has better style?

ICA

A

```
// variable to control fence animation
int position = 0;

void draw() {
  background(200, 230, 255);
  strokeWeight(0);
  fill(100, 255, 100);
  rect(0, 140, 300, 100);
  strokeWeight(4);

  // repeatedly draw the fence posts
  for(int i = 0; i < 2001; i += 25) {
    line(position + i, 50, position + i, 150);
  }

  line(0, 75, 300, 75);
  line(0, 125, 300, 125);
  // increment to control fence movement speed
  position = position - 1;
}
```

B

```
int P1z = 0;
void  draw()  {  background(200, 230, 255);
      strokeWeight(0);
fill(100, 255, 100);
    rect(0, 140, 300, 100); strokeWeight(4);
      for(int i = 0;
i < 2001; i += 25)
 {
line(P1z + i, 50,   P1z + i, 150);
   }
  line(0, 75, 300  , 75);
      line( 0 ,   125, 300, 125);  P1z = P1z - 1;
   // What is going on here?
}
```

ICA

Question 4. Name at least two things that you identified as bad style.

What makes good style

- Indentation
- Commenting/Documentation
- Naming
- Spacing

Indentation

- We've seen several types of code-constructs that use curly-braces
  - setup and draw (and other user defined functions)
  - for-loops (and other structures like if-statements)

- Whenever a new "level" of curly-braces is reached, best-practice is to indent the code accordingly
- In Processing, indentation is 2-spaces
  - Varies between language!

Indentation

```
void draw () {

background(100, 200, 250);
       strokeWeight(5);
 fill(200, 100, 50);
    rect(50, 50, 100, 100);
}
```

Indentation

One pairing of open/close curly-braces indicates a new "chunk" of code

```
void draw () {
    background(100, 200, 250);
    strokeWeight(5);
    fill(200, 100, 50);
    rect(50, 50, 100, 100);
}
```

Indentation

One pairing of open/close curly-braces indicates a new "chunk" of code

```
void draw () {
    background(100, 200, 250);
    strokeWeight(5);
    fill(200, 100, 50);
    rect(50, 50, 100, 100);
}
```

Because of this, indent all code in-between 1 level (which is 4-spaces or 1-tab)

Indentation - Let’s fix this code

```
void draw () {background(100, 200, 250);
fill(0, 0, 255);
     if (mousePressed) {
    for (int i = 0; i < 10 ; i += 1) {
  rect(i*20, 20, 15, 15);
    }
fill(255, 0, 0);
  }   rect(50, 50, 100, 100);
}
```

Indentation

Indented 1st level

Indented 2nd level

Indented 3rd level

Python - Wrong indentation = Wrong Program

Syntax Error:

```
if 5 > 2:
print("Five is greater than two!")
```

Syntax Error:

```
if 5 > 2:
 print("Five is greater than two!")
        print("Five is greater than two!")
```

Commenting and Documenting

- Comments should be used in multiple ways
  - High-level descriptions of a whole program, or function
    - Typically at top of program or function
  - Fine-grained descriptions of what a specific line of code does
    - Typically above or to the right of a line of code

Commenting and Documenting

- As programs grow in size, it can become harder to keep track of what the program is doing and how it works
- Programmers use **comments** to make code easier to understand
- There are two types of comments in processing:
  - One-line comments
  - Block comments
- You can write your Algorithm in plain English in comments and then translate these steps into Processing code (TODO keyword)

// In-line Comments

- Explain lines that might be difficult to understand
  - No need to comment lines that we are already familiar with. For example, size.
- Describe the variables
- Describe blocks of code
- Comment code that it’s not easy to understand or took you some time to write

Can you understand this code without running it?

```
int offset = 0;



void setup() {
  size(300, 200);
}

void draw() {
  background(200, 230, 255);
  strokeWeight(0);
  fill(100, 255, 100);
  rect(0, 140, 300, 100);
```

```
  strokeWeight(4);
  for(int i = 0; i < 1001; i += 25) {
    line(offset + i, 50, offset + i, 150);
  }

  line(0, 75, 300, 75);
  line(0, 125, 300, 125);

  offset = offset - 1;
}
```

Can you understand this code without running it? - Useful comments

```
int offset = 0; //displacement along the
X-axis

void setup() {
  size(300, 200);
}

void draw() {
  background(200, 230, 255); //light blue
  strokeWeight(0);
  fill(100, 255, 100); //light green
  rect(0, 140, 300, 100);  //grass
```

```
  //horizontal poles of the fence
  strokeWeight(4);
  for(int i = 0; i < 1001; i += 25) {
    line(offset + i, 50, offset + i, 150);
  }
  //fence rails
  line(0, 75, 300, 75);
  line(0, 125, 300, 125);

  offset = offset - 1; //to move all points to the
left
}
```

Can you understand this code without running it? - Not useful comments

```
int offset = 0; //declare variable

void setup() {
  size(300, 200); //call function size
}

void draw() {
  background(200, 230, 255); //background
  strokeWeight(0);//stroke
  fill(100, 255, 100); //fill
  rect(0, 140, 300, 100);  //paint a rec
```

```
  strokeWeight(4);//stroke
  //for loop
  for(int i = 0; i < 1001; i += 25) {
    line(offset + i, 50, offset + i, 150);//line
  }
  line(0, 75, 300, 75); //draws a line
  line(0, 125, 300, 125); //draws a line

  offset = offset - 1; //subtract 1 from offset
}
```

/* Block Comments */

```
/*
 * Name: J Student
 * Description: Animated fence using for loops
 * Algorithm:
 *  1. Paint background
 *  2. Paint fence for frame 1
 *  3. Paint fence one pixel to the left for frame 2
 *  4. Paint fence one pixel to the left for frame 3
 *  5. Continue moving the fence for each frame
 */
```

```
/*
 * Name: J Student
 * Description: Draws two rows of shapes
 */

void setup() {
  // Don't want to use the default of 60 - I use 30 instead
  frameRate(30);
}


void draw() {
  fill(100, 200, 100);  //green
  // draws one row of squares and another row of circles below it
  for(int i = 0; i < 300; i += 50) {
    rect(i, 50, 40, 40);
    ellipse(i-25, 100, 40, 40);
  }
}
```

Spacing

- There should be a space between the various “chunks” of code in your program
  - Between **setup** and **draw**
    - And other functions . . .
  - Between chunks of code that do logically different things
  - Between global variable declarations and functions

ICA

What does this code do?

```
int P1z = 0;


void setup () { size(300, 200); }
void  draw()  {  background(200, 230, 255);
      strokeWeight(0);
fill(100, 255, 100);
    rect(0, 140, 300, 100); strokeWeight(4);
      for(int i = 0;
i < 2001; i += 25)
 {
line(P1z + i, 50,   P1z + i, 150);
   }
  line(0, 75, 300  , 75);
      line( 0 ,   125, 300, 125);  P1z = P1z - 1;
   // What is going on here?
}
```

How about this?

```
// variable to control
// the fence animation
int position = 0;

/**
 * Set the size of the
 * Fence animation
 */
void setup () {
  size(300, 200);
}
```

ICA

```
void draw() {
  background(200, 230, 255);
  strokeWeight(0);
  fill(100, 255, 100);
  rect(0, 140, 300, 100);
  strokeWeight(4);

  // repeatedly draw the fence posts
  for(int i = 0; i < 2001; i += 25) {
    line(position + i, 50, position + i, 150);
  }

  line(0, 75, 300, 75);
  line(0, 125, 300, 125);
  // increment to control fence movement speed
  position = position - 1;
}
```

Processing-Enforced naming rules

- There are some processing-enforced rules for variables names
  - Can start with any letter (A-Z, a-z), underscore (_), dollar sign ($)
  - Can continue with any of the above, and can also continue with digits (0-9)
  - Can have unlimited length
  - Cannot be any Processing keyword (void, if, else, for, false, true, null, etc)

Naming Style Guidelines - camelCase naming

- In addition to the enforced rules, programmers have come up with ***best-practices*** for variable names
- Some programming languages have differing naming guidelines
- In processing, we use ***camelCase***
- **camelCase** - compounding words together where the first word is not capitalized, and the rest are, and no spaces
  - iPhone eBay

Question 5

- Go to the class website
  - Download square_example.pde
  - Modify it to use good variable names (if there are any), correct indentation and spacing, and comments

square_example.pde

```
void setup() { size(200, 200) ;} void draw(){
background(100, 200, 250);
fill(0, 0, 255);
if (mousePressed) {
if (mouseX > 100) {
fill(255, 0, 0);
}}
if (    mouseButton ==  RIGHT) {
background(0, 0, 0  );
strokeWeight(7);
}                    rect(50, 50, 100, 100  ) ;}
```

https://app.tophat.com/e/170052

ICA

Which of these variables are valid?

```
(1)  int 123abc = 17;
(2)  int LARGE NUMBER = 10000;
(3)  float AnotherNumber34 = 123.456;
(4)  char not?sure_ = ‘r’;
(5)  int one_TWO_three = 123;
(6)  char 3verything3 = 7;
(7)  int true = 1
```

Variable naming - camelCase

- What would the following convert to in camelCase? Answer in the ICA handout.
  - “a very large number”
  - “the 3rd best item”
  - “times 10”
  - “red, green, blue”
  - “6 afraid of 7”

Variable naming - camelCase

- What would the following convert to in camelCase?
  - “a very large number” - aVeryLargeNumber
  - “the 3rd best item” - the3rdBestItem
  - “times 10” - timesTen
  - “red, green, blue” - redGreenBlue
  - “6 afraid of 7” - sixAfraidOfSeven

ICA

Variable naming

For each variable, determine if it is either **(A)** an error, **(B)** valid, but not good style, or **(C)** valid and good style

```
(1)  int 8timesTheForce = 16;
(2)  int tallestPerson = 82;
(3)  float InchesToFeet = 12.0;
(4)  char FIRSTCharacter = ‘A’;
(5)  bool skyIsBlue = true;
(6)  int 7eight9 = 789;
```

Variable naming

For each variable, determine if it is either **(A)** an error, **(B)** valid, but not good style, or **(C)** valid and good style

```
(1)  int 8timesTheForce = 16;     A
(2)  int tallestPerson = 82;      C
(3)  float InchesToFeet = 12.0;   B
(4)  char FIRSTCharacter = ‘A’;   B
(5)  bool skyIsBlue = true;       C
(6)  int 7eight9 = 789;       A
```

Which has better style?

ICA

```
// variable to control fence animation
int position = 0;

void draw() {
  background(200, 230, 255);
  strokeWeight(0);
  fill(100, 255, 100);
  rect(0, 140, 300, 100);
  strokeWeight(4);

  // repeatedly draw the fence posts
  for(int i = 0; i < 2001; i += 25) {
    line(position + i, 50, position + i, 150);
  }

  line(0, 75, 300, 75);
  line(0, 125, 300, 125);
  // increment to control fence movement speed
  position = position - 1;
}
```

```
int P1z = 0;

void setup () { size(300, 200); }
void  draw()  {  background(200, 230, 255);
      strokeWeight(0);
fill(100, 255, 100);
    rect(0, 140, 300, 100); strokeWeight(4);
      for(int i = 0;
i < 2001; i += 25)
 {
line(P1z + i, 50,   P1z + i, 150);
   }
  line(0, 75, 300  , 75);
      line( 0 ,   125, 300, 125);  P1z = P1z - 1;
   // What is going on here?
}
```