Computer Creativity

Final Class !

Slides courtesy of Dr. Abdallah Mohamed.

COSC 123 Final Exam

- Check SSC for the official date and time of the Final Exam
- There will be some multiple choice questions, but the majority will be coding tasks
- The final exam will be:
  - Cumulative
  - Live (2.5 hours), invigilated, but no proctoring.
  - Open book, open-notes, open-web but no cheating sites like Chegg/Course-Hero/Bartleby etc
  - IDEs are ok
  - On Canvas, using Gradescope and GitHub

COSC 123 Final Exam

- Thursday Apr 29 at 15:30
- Online
- **Time:** 2.5 hours
- The exam covers **all course material** as indicated in the syllabus and during the lectures.
  - You will NOT be formally tested on Git and Command Line, but you will need those concepts to do the final exam (i.e. accept a GitHub Classroom repository)

Gradescope Demo

DISTINGUISHED SPEAKER SERIES

From Big Data to Your Data: How Data Driven Technologies are shaping the future with Nora Young.

Tuesday April 13, 2021

Register here

Student Evaluations of Teaching (SEoT)

- You may have received an email that student evaluations of teaching is now open for this course.

- Research shows that SEoT are flawed because they are influenced by unconscious and unintentional biases.

Student Evaluations of Teaching (SEoT)

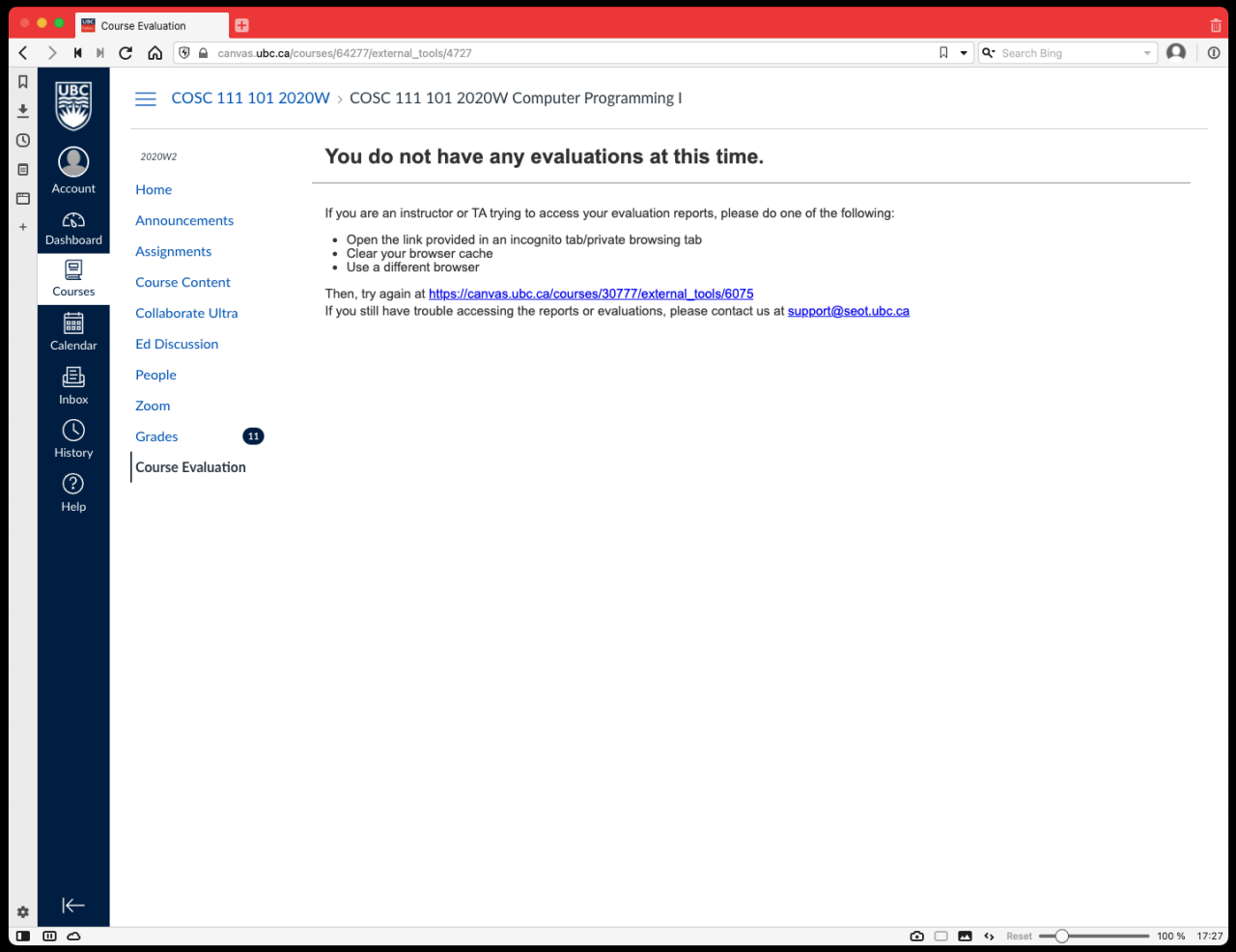
- Despite their flaws, Teaching Evaluations are used to departments to:
  - Make decisions on Tenure and Promotion
  - Decide which courses instructors teach
  - Rate/rank grant applications and awards

- More important to me however, is how you felt about the course content, the structure, and me as an instructor.

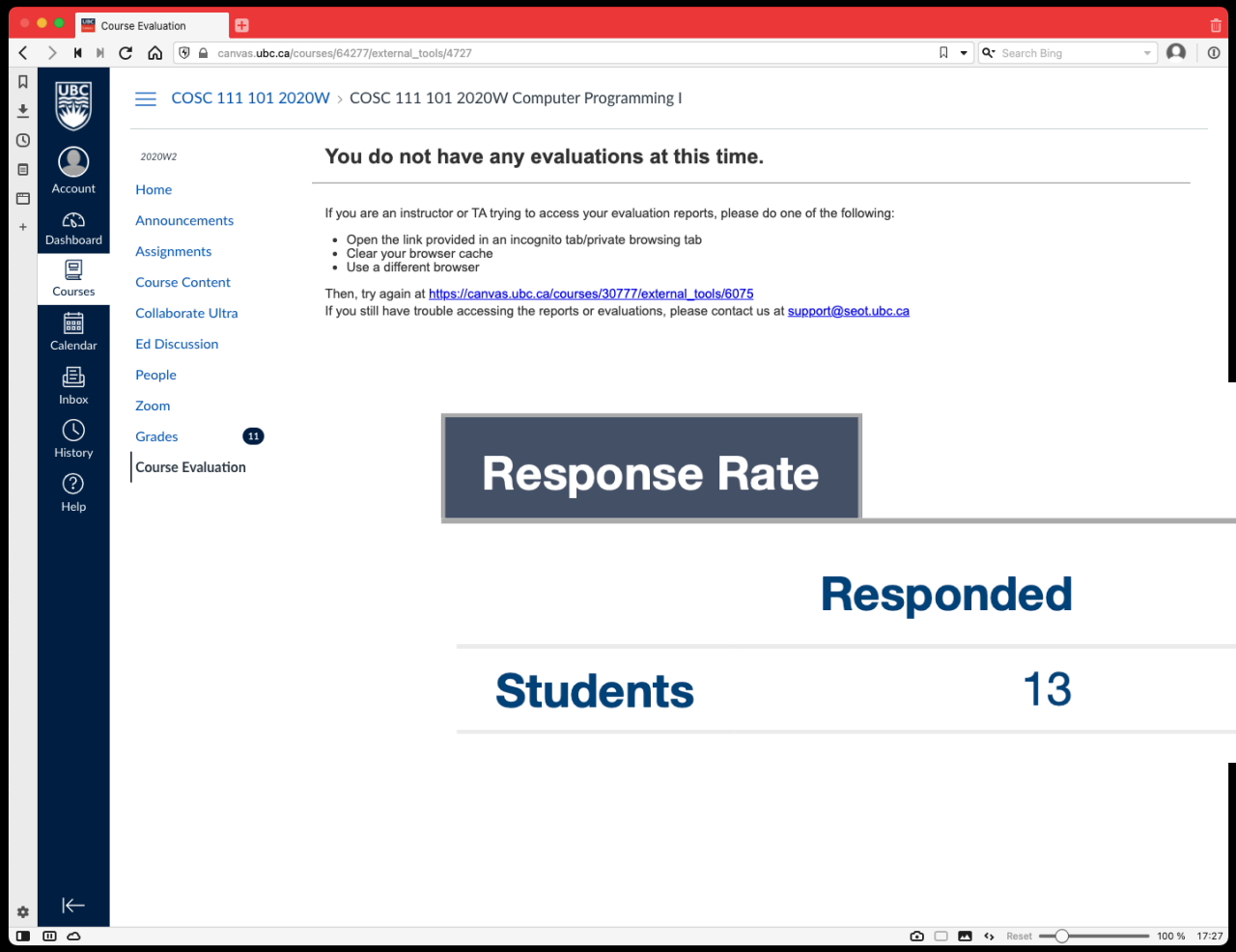
  I want to hear from you!

- My goal is to get at least a 70% response rate on SeOT, the more the merrier!

Student Evaluations of Teaching (SeOT)



Student Evaluations of Teaching (SeOT)



Response Rate

	Responded	Invited	% Rate
Students	13	133	9.77%

What you have learned so far…

Overview

Items include:

- Components of Processing (PDE + API (libraries) + syntax (Java))
- Printing to the console
- Drawing primitive shapes and text
- Color
- Shape coordinates (origin)
- The coordinate system + how to transform it
- User and System Variables

Programming Basics

- Introduction:
  - Algorithms and their properties
  - Programming, program, language
  - The five basic steps in software development
- Error types
- Programming modes
  - Java, JavaScript, Python, Android, etc.
- Components of Processing
  - PDE + API (libraries) + syntax (Java)
- Setting the background and the sketch size
  - `background`, `size`
- Printing to the console
  - `print`, `println`

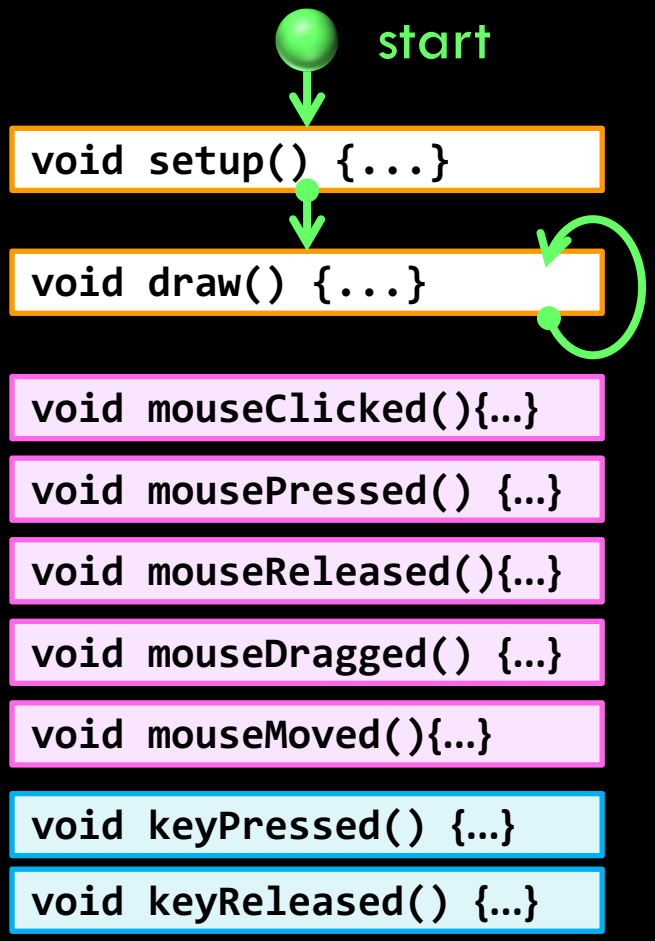
Drawing & Coloring

- Drawing primitive shapes
  - `point()`, `line()`, `rect()`, `ellipse()`, `triangle()`, `quad()`, `Bezier()`
- Shape coordinates (origin)
  - `rectMode()`, `ellipseMode()`
- Filling and stroke attributes
  - `fill()`, `stroke()`, `strokeWeight()`, `noFill()`, `noStroke()`
- Adding Text
  - `text()`, `fill()`, `textSize()`
- Color
  - RGB, HSB: `colorMode()`
  - Controlling transparency
- The coordinate system + how to transform it
  - `translate()`, `rotate()`, `scale()`
  - `pushMatrix()`, `popMatrix()`
  - Order matters!

Where to Write Your Code

- **Static mode**
  - monolithic program, runs once, with not functions
- **Active mode**
  - `Setup()` and `draw()`, and where to put each statement
  - Controlling the `frameRate()`
  - Stopping an animation using `noLoop()`
- *Event-based functions*

  mouseClicked(), mousePressed(), …, keyPressed(), keyReleased()
  - How to know which key is pressed?



Using Variables

- Value, variable, and location
- Variable types:

  `byte`, `short`, `int`, `long`, `float`, `double`, `char`, `Boolean`

  `String`, `color`
- Using `final` for declaring constants
- Naming rules and guidelines (for variable and constants)
- Math operators and expressions
  - *Binary operators:* `+, -, *, /, %`
  - *Unary operators:* `-3, x++, y--`
  - *Augmented assignment:* `-=, +=, /=, *=, %=`
- Variable scope
  - Local vs global variables

How to Animate "Things"

(1) Identify which attributes you want to animate (e.g. size, color, location, etc.)

(2) for each attribute you want to animate, create and initialize a global variable.

(3) In `draw()`, use the global variables to represent the attributes.

(4) Change the value of your global variables either:

- ***in `draw()`***
  - For continuous animation (e.g. falling rain drops).
- ***in an event-based function (e.g. `keyPressed()`)***
  - For interactive animations (e.g. controlling character position with keyboard)

System Variables & Helpful Functions

- System variables

  Mouse location: `mouseX`, `mouseY`, `pmouseX`, `pmouseY`

  Dimentsions: `width`, `height`, `displayWidth`, `displayHeight`

  Frames: `frameCount`, `frameRate`

  Events: `keyPressed`, `key`, `keyCode`, `mousePressed`, `mouseButton`

- Math functions:

  `abs()`, `round()`, `floor()`, `ceil()`, `pow()`, `sq()`, `sqrt()`, `max()`, `min()`, `dist()`

  `sin()`, `cos()`, `tan()`, `asin()`, `acos()`, `atan()`, `degrees()`, `radians()`

- Time functions:

  `second()`, `minute()`, `hour()`, `day()`, `month()`, `year()`, `millis()`

Useful Functions and Images

- Useful functions
  - Math (`sin`,`abs`,…), casting(`int`, `float`), `noLoop`, `loop`, …
  - Randomness: `random`, `noise`
  - Range: `map`, `norm`, `constrain`
- Images:
  - `PImage`, `loadImage`, `image`, `imageMode`, `width`, `height`

Control Structures, Custom Functions

- Conditionals
  - Basics: `if-else`, `switch`, operators (`>`, `==`,..., `&&`, `||`,..), `equals`
  - Ideas:
    - Deciding based on system variable (color-mouseX, stop-frameCount)
    - Controlling items with keyboard
    - Deciding based on object state: e.g. buttons (clickable, toggle)
    - Bouncing attributes
    - Physics 101: gravity
- Loops: `while`, `for`, `++`, `--`, common problems
- Functions
  - Divide-and-conquer, game loops, animations with multiple scenes

OOP and Arrays

- Basics of OOP (Object Oriented Programming) in Processing
  - Defining classes, objects, methods, and attributes (instance variables).
  - Component of a class
  - Creating objects from classes using `new`.
  - Constructors
  - OOP Thinking – using objects in animations
- Arrays
  - Basics of arrays
    - structure, indexing, and bound checking
    - Creating and initializing arrays.
    - Using `for` loops to process array elements
- Arrays of objects
  - Concepts
  - Creating and initializing

Sample Questions

Some questions you should try to do as part of your review process.

Question

What is the value?

How many attributes do we have?

A. 1

B. 3

C. 4

D. 8

E. 0

```
class Button{
  float x, y, r;   boolean active = false;
  Button(){x=50; y=50; r=40;}
  Button(float x1, float y, float r1){x=x1; y=y1; r=r1;}
  void display(){
    strokeWeight(5);
    textSize(25);   textAlign(CENTER,CENTER);
    if(active) {
      fill(0, 200, 0);  stroke(0,255,0);
      ellipse(x,y,2*r,2*r);
      fill(200,255,200);text("ON",x,y);
    }else{
      fill(180, 0, 0);  stroke(255,0,0);
      ellipse(x,y,2*r,2*r);
      fill(100,0,0);    text("OFF",x,y);
    }
  }
  void checkClicked(){
   if(dist(mouseX,mouseY,x,y)<r) active=!active;
  }
}
```

Question

What is the value?

How many constructors do we have?

A. 0

B. 1

C. 2

D. 3

E. 4

```
class Button{
  float x, y, r;   boolean active = false;
  Button(){x=50; y=50; r=40;}
  Button(float x1, float y, float r1){x=x1; y=y1; r=r1;}
  void display(){
    strokeWeight(5);
    textSize(25);   textAlign(CENTER,CENTER);
    if(active) {
      fill(0, 200, 0);  stroke(0,255,0);
      ellipse(x,y,2*r,2*r);
      fill(200,255,200);text("ON",x,y);
    }else{
      fill(180, 0, 0);  stroke(255,0,0);
      ellipse(x,y,2*r,2*r);
      fill(100,0,0);    text("OFF",x,y);
    }
  }
  void checkClicked(){
   if(dist(mouseX,mouseY,x,y)<r) active=!active;
  }
}
```

Question

What is the value?

What is b.x after

Button b = new Button();

A. 0

B. 40

C. 50

D. Unknown

E. error

```
class Button{
  float x, y, r;   boolean active = false;
  Button(){x=50; y=50; r=40;}
  Button(float x1, float y, float r1){x=x1; y=y1; r=r1;}
  void display(){
    strokeWeight(5);
    textSize(25);   textAlign(CENTER,CENTER);
    if(active) {
      fill(0, 200, 0);  stroke(0,255,0);
      ellipse(x,y,2*r,2*r);
      fill(200,255,200);text("ON",x,y);
    }else{
      fill(180, 0, 0);  stroke(255,0,0);
      ellipse(x,y,2*r,2*r);
      fill(100,0,0);    text("OFF",x,y);
    }
  }
  void checkClicked(){
   if(dist(mouseX,mouseY,x,y)<r) active=!active;
  }
}
```

Question

What is the value?

How many **Button objects** do we have after running this statement:

Button[] b = new Button[5];

A. 0

B. 4

C. 5

D. 6

E. error

```
class Button{
  float x, y, r;   boolean active = false;
  Button(){x=50; y=50; r=40;}
  Button(float x1, float y, float r1){x=x1; y=y1; r=r1;}
  void display(){
    strokeWeight(5);
    textSize(25);   textAlign(CENTER,CENTER);
    if(active) {
      fill(0, 200, 0);  stroke(0,255,0);
      ellipse(x,y,2*r,2*r);
      fill(200,255,200);text("ON",x,y);
    }else{
      fill(180, 0, 0);  stroke(255,0,0);
      ellipse(x,y,2*r,2*r);
      fill(100,0,0);    text("OFF",x,y);
    }
  }
  void checkClicked(){
   if(dist(mouseX,mouseY,x,y)<r) active=!active;
  }
}
```

Question

Objects and Object References

How many objects are created by this code?

```
Ball a, b, c;

a = new Ball();
c = a;
b = new Ball();
```

A. 1

B. 2

C. 3

D. 4

Question

Objects and Object References

What is the radius of the ball referenced by `d`?

```
Ball a, b, c, d;

a = new Ball(50);    // radius = 50
c = a;
b = new Ball(100);   // radius = 100
a = b;
d = c;
```

A. unknown

B. 50

C. 100

D. undefined

Question

Objects and Object References

How much money is in the account referenced by the `d`?

```
Ball a, b, c, d;

b = new Ball(50);
c = b;
a = new Ball(100);
b = a;
d = c;
```

A. unknown

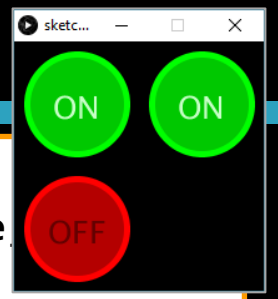
B. 50

C. 100

D. undefined

Previously…

OOP Toggle Buttons (3 buttons)



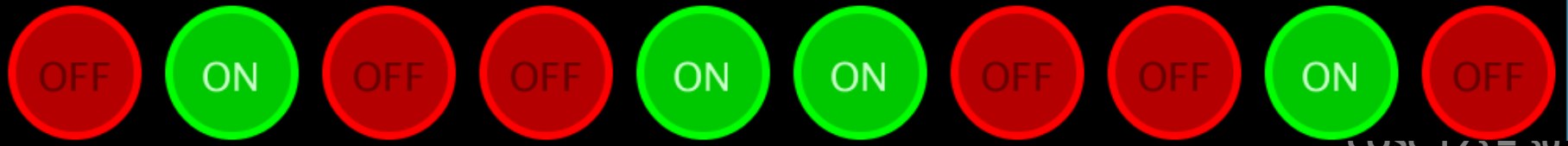

```
Button b1, b2, b3;
void setup() {
  size(200, 200);
  b1 = new Button();
  b2 = new Button();
  b2.x = 150;
  b3 = new Button();
  b3.y = 150;
}
void draw() {
 background(0);
 b1.display();
 b2.display();
 b3.display();
}
void mousePressed(){
 b1.checkClicked();
 b2.checkClicked();
 b3.checkClicked();
}
```

```
class Button{
  float x, y, r;   boolean active = false;
  Button(){x=50; y=50; r=40;}
  Button(float x1, float y, float r1){x=x1; y=y1; r=r1;}
  void display(){
    strokeWeight(5);
    textSize(25);   textAlign(CENTER,CENTER);
    if(active) {
      fill(0, 200, 0);  stroke(0,255,0);
      ellipse(x,y,2*r,2*r);
      fill(200,255,200);text("ON",x,y);
    }else{
      fill(180, 0, 0);  stroke(255,0,0);
      ellipse(x,y,2*r,2*r);
      fill(100,0,0);    text("OFF",x,y);
    }
  }
  void checkClicked(){
   if(dist(mouseX,mouseY,x,y)<r) active=!active;
  }
}
```

Array of Buttons

- Create an array of 10 Buttons as shown below

```
int N = 10;
Button[] b = new Button[N];
void setup() {
  size(1000, 100);
  for(int i=0; i<N; i++)
    b[i] = new Button(100*(i+1)-50,50,40);
}
void draw() {
  background(0);
  for(int i=0; i<N; i++)
    b[i].display();
}
void mousePressed(){
  for(int i=0; i<N; i++)
    b[i].checkClicked();
}
```



Question

Use of constrain()

We want background to gradually change from black to white. Which code is better?

(1)

```
int shade = 0;
void draw(){
  background(shade);
  shade++;
}
```

(2)

```
int shade = 0;
void draw(){
  background(shade);
  shade = constrain(shade+1,0,255);
}
```

A. (1) is better than (2)

B. (2) is better than (1)

C. They are both the same

D. I don't understand what you are talking about.

Controlling Image Opacity

Which of the following is a statement that we can use to control set the transparency of an image to 50%?

A. `tint(255);`

B. `tint(128);`

C. `tint(255,128)`

D. `opacity(128);`

E. `transparency(50)`

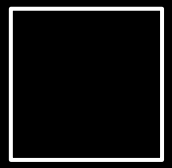
Making Decisions

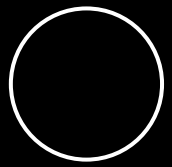
What is the output of this code?

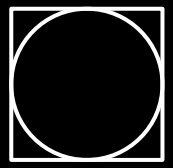
```
noFill();  rectMode(CENTER); stroke(255);

int num = 9;

if (num == 10)
  rect(50,50,50,50);
  ellipse(50,50,50,50);
```

A. 

B. 

C. 

D. Something else

Question

Boolean Expressions

What is the output of this code?

```
int x = 10, y = 20;
if (x >= 5) {
  print("bigx");
  if (y >= 10)
     print("bigy");
} else if (x == 10 || y == 15)
  if (x < y && x != y)
     print("not equal");
```

A. `bigx`

B. `bigy`

C. `bigxnot equal`

D. `bigxbigynot equal`

E. `bigxbigy`

Question

Switch Statement

What is the output of this code?

```
int num=2;
switch (num){
  case 1: print("one");
  case 3: print("three");     break;
  case 2: print("two");
  default:print("other");     break;
}
```

A. three

B. two

C. twothree

D. twoother

E. other

Loops vs. draw()

Which code is better? We want the ball to gradually move from left to right.

(1)

```
int x = 0;
void draw(){
  background(0);
  ellipse(x,50,20,20);
  x++;
}
```

(2)

```
for(int x = 0; x < 100; x++){
  background(0);
  ellipse(x,50,20,20);
}
```

A. (1) is better than (2)

B. (2) is better than (1)

C. They are both the same

D. I don't understand what you are talking about.

Functions

What is the output of this code?

```
int subtractNum(int a, int b) {
 return a-b;
}
void draw() {
 int x=5, y=8;
 int result = subtractNum(x, y);
 print(result + subtractNum(y, x));
 noLoop();
}
```

A. error

B. 3

C. -3

D. 0

Photo by **Giftpundits.com** from **Pexels**

What you should look at before COSC 121...