

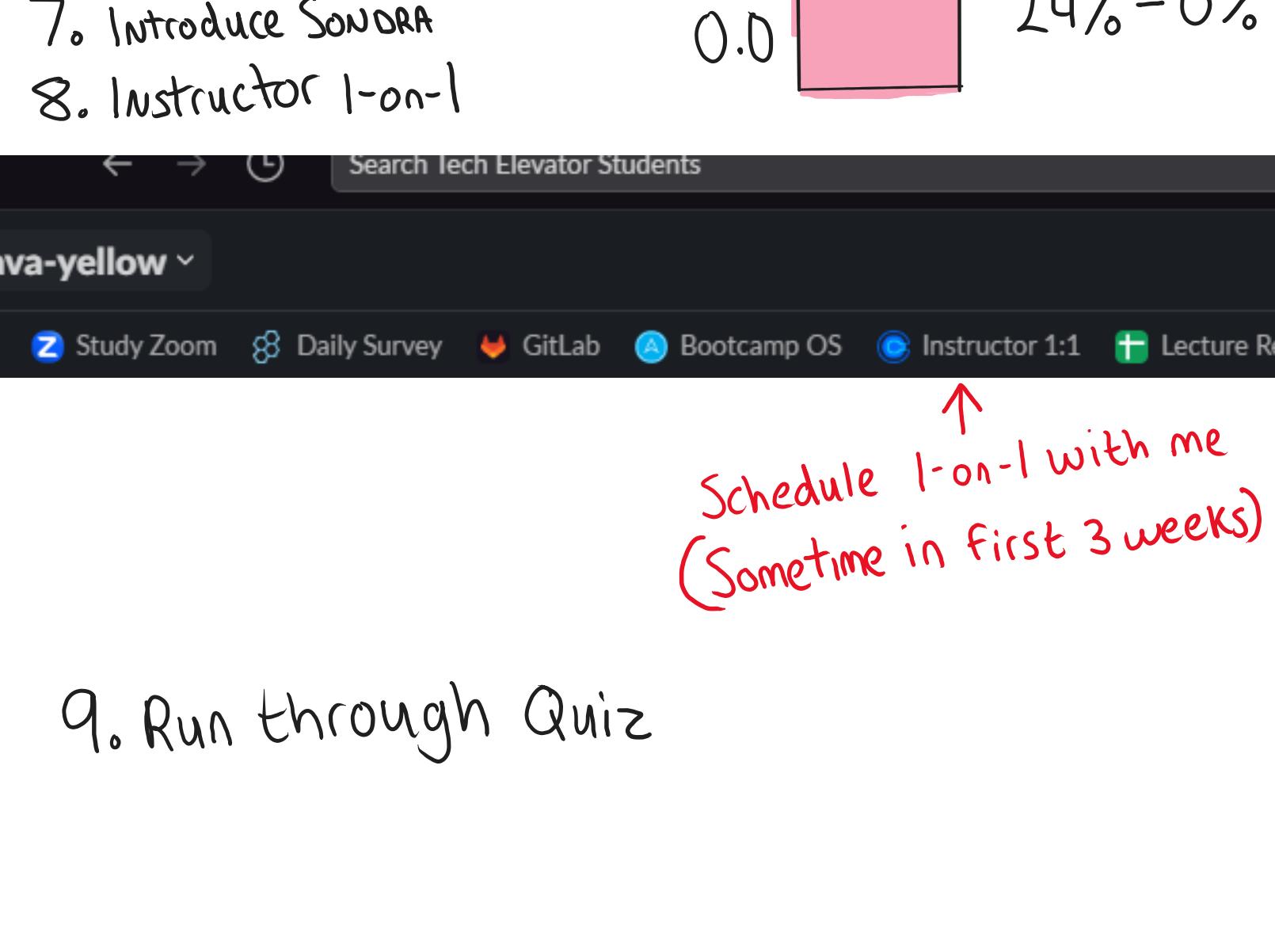
PLEASE COMPLETE THE SOCRATIVE Pulse Survey

URL: gosocrative.com

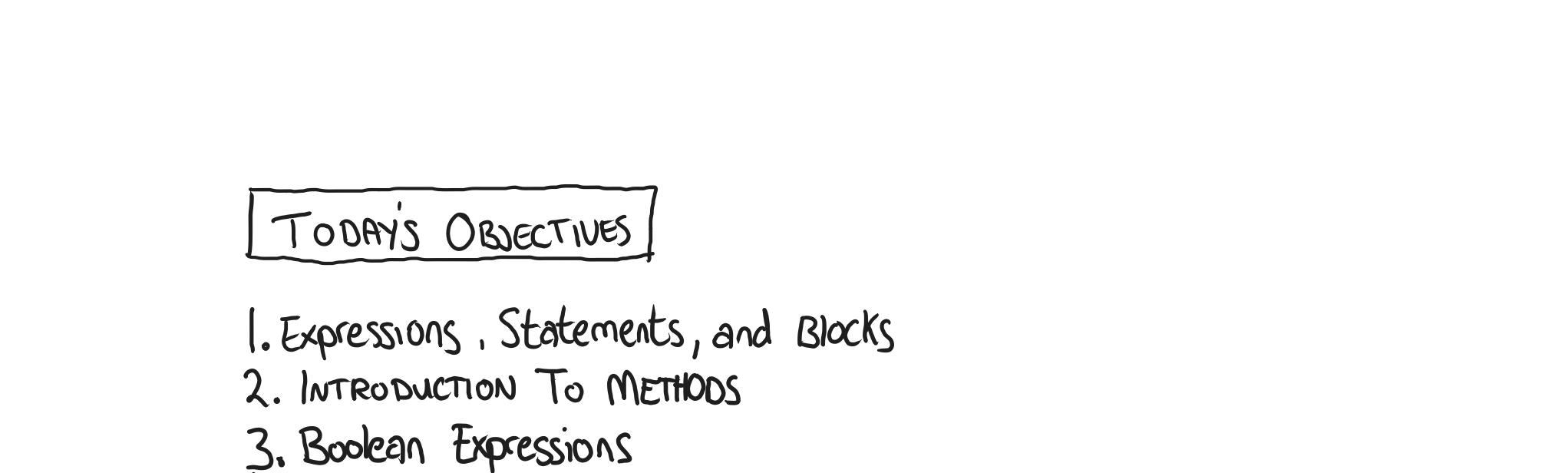
ROOM NAME: JAVAGOLD

```
#50
int firstVariable;
int secondVariable;
double sum = (double)firstVar...;
```

GRADING



double roomsByBill = 1/2.15;
double roomsByJill = 1/1.90;



9. Run through Quiz

TODAY'S OBJECTIVES

1. Expressions, Statements, and Blocks
2. Introduction To METHODS
3. Boolean Expressions
4. Conditional Statements

Expressions, Statements, and Blocks

An expression is not a complete thought, instead it must first be solved (evaluated), but when solved it has a single answer.

Examples: $5 - 3$

$$x - y$$

A statement is a complete thought. It does not need to be solved. A statement is roughly equivalent to a sentence in natural language.

Examples: $S - 3 = 2$

```
int x = 5 - 3;
double area = length * height;
```

A block is a group of statements that need to be executed as a single unit.
IN JAVA, blocks are identified by curly braces { }

Examples:

```
{ int currentFloor = 1;
    currentFloor = 2;
}
```

METHODS

A method is a reusable, named block of code.

Methods are defined by a method signature that identifies:

1. who CAN USE IT (ACCESS MODIFIER)
2. WHAT IT WILL RETURN (RETURN TYPE)
3. HOW TO CALL IT (NAME)
4. WHAT INPUT IT TAKES (PARAMETERS)

```
public int addNumbers int x, int y) {
    int sum = x + y;
    return sum;
}
```

METHOD SIGNATURE

accessor	return type	name	(parameters)
public	void	addNumbers	(int paramOne, String paramTwo)
private	int	printMenu	
	double		

accessor - identifies who can use it (public, private, protected...)
return type - type of data the method will return (int, String, float, void...)
name - descriptive name used to call the method
parameters - list of variables that must be populated to call the method

BOOLEAN EXPRESSIONS

An expression that evaluates to either true or false.

boolean isRed = False;

if:

```
if (boolean expression){
```

 code to run when true

}

:

if-else:

```
if (boolean expression){
```

 code to run when true

}

:

```
else{
```

 code to run when false

}

:

COMPARISON OPERATORS

OPERATORS	MEANING
$==$	Equal to
$!=$	Not equal to
$>$	Greater than
$<$	Less than
\geq	Greater than or Equal to
\leq	Less than or Equal to

LOGICAL OPERATORS

OPERATORS	MEANING
$\&\&$	AND (Both conditions must be true)
$\ $	OR (At least one condition must be true)
\wedge	XOR (Both conditions must NOT BE THE SAME)
!	NOT (switches the result)

Logical Operator Truth TABLE

A	B	! A	A && B	A B	A ^ B
True	True	False			
True	False	False			
False	True	True			
False	False	True			