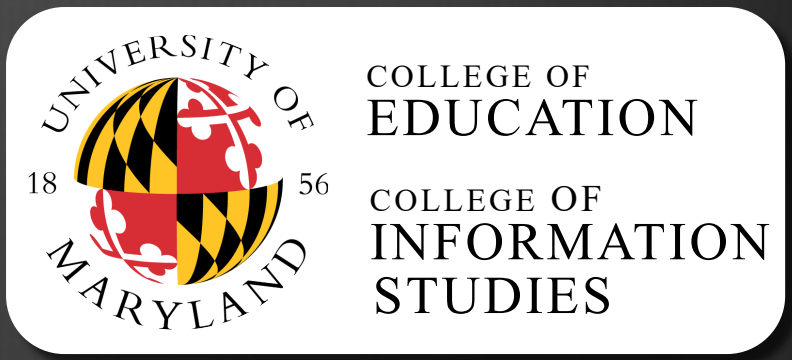
# The Role of Block-based Programming in Computer Science Education

David Weintrop
weintrop@umd.edu
University of Maryland

Raspberry Pi CER Seminar
December 1 , 2020



*The focus of my work is the creation of meaningful, accessible, and equitable computational learning experiences for all learners.*

# Block-based Programming

# Block-based Programming

PENCIL code

forever
imagine
program
share

SCRATCH

Welcome to MIT App Inventor

Snap!

CODE

App Lab

Blockly

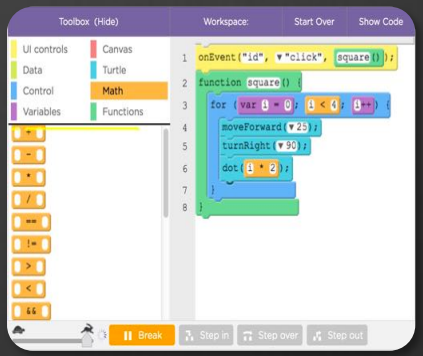
# New CS Curricula using Block-based Programming

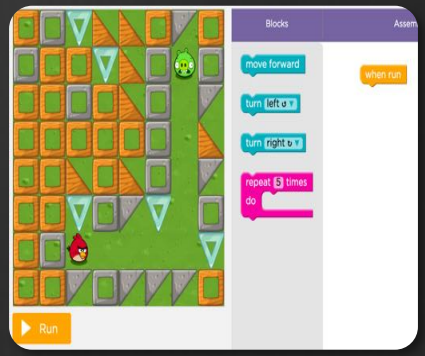
Exploring Computer Science

Beauty and Joy of Computing

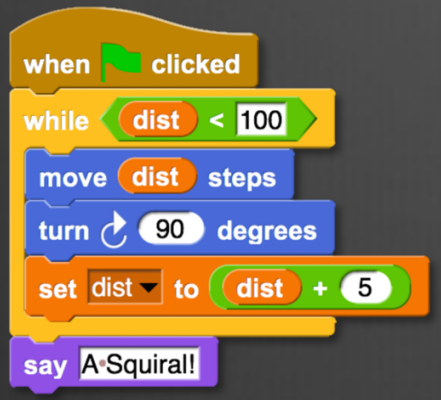
An AP CS: Principles course.

C O
D E





# Block-based Programming

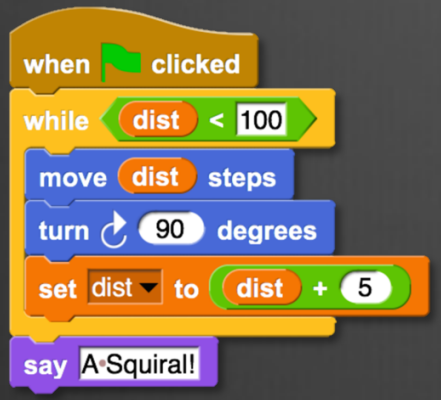



Block-based Environments

```
public void squiral() {
    Turtle turtle = new Turtle();
    int dist = 0;
    while (dist < 100) {
        turtle.forward(dist);
        turtle.right(90);
        dist = dist + 10;
    }
    turtle.say("A Squiral!");
}
```

Text-based Languages

# Research Questions




Block-based Environments

VS

```
public void squiral() {
    Turtle turtle = new Turtle();
    int dist = 0;
    while (dist < 100) {
        turtle.forward(dist);
        turtle.right(90);
        dist = dist + 10;
    }
    turtle.say("A Squiral!");
}
```

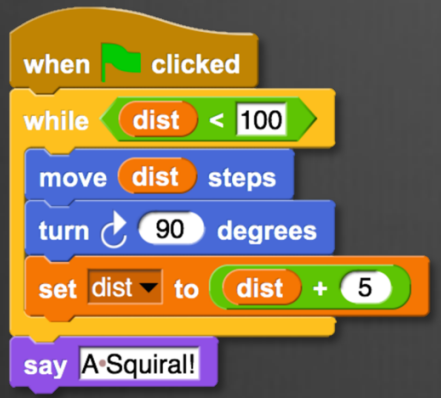
Text-based Languages

## The Modality Set

How does programing modality affect:

- Emerging student understandings
- Student attitudes and perceptions
- Student programming practices

# Research Questions




Block-based Environments

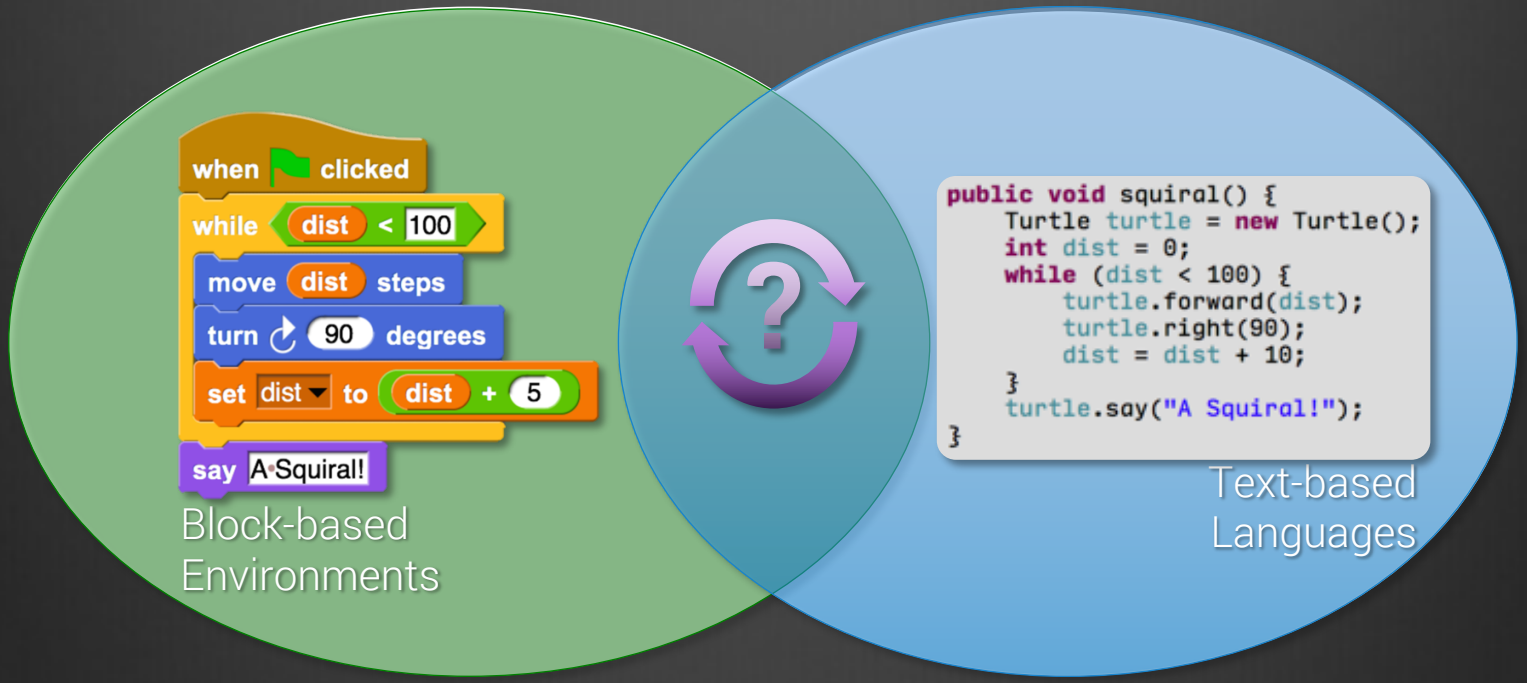
```
public void squiral() {
    Turtle turtle = new Turtle();
    int dist = 0;
    while (dist < 100) {
        turtle.forward(dist);
        turtle.right(90);
        dist = dist + 10;
    }
    turtle.say("A Squiral!");
}
```

Text-based Languages

## The Transition Set

Do understandings and practices developed in introductory tools persist after transitioning to professional programming languages and modalities?

# Research Questions




## The Design Set

Is there a “best of both worlds”?
What does it look like?

# *Methods & Participants*

# Study Design

Classroom-based, 3-condition, 2-phase, Quasi-Experimental, Mixed Method Study

3 sections of programming elective, 1 Teacher, ~30 students/class

Public, selective enrollment high school in Midwestern city

41% White, 27% Hispanic, 11% Asian, 11% Multiracial, 10% Black

75 male students; 15 female students

47%  speak a language other than English in their households

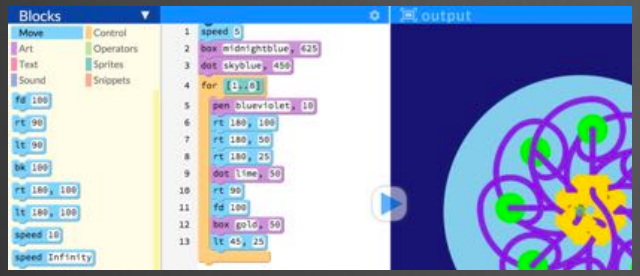
58.6% (school wide) come from economically disadvantaged households

# Study Design

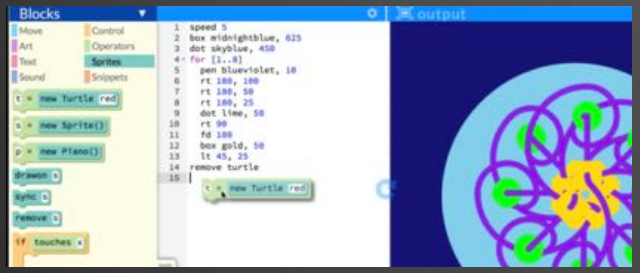
Classroom-based, 3-condition, 2-phase, Quasi-Experimental, Mixed Method Study

## 3 Conditions

Block-based

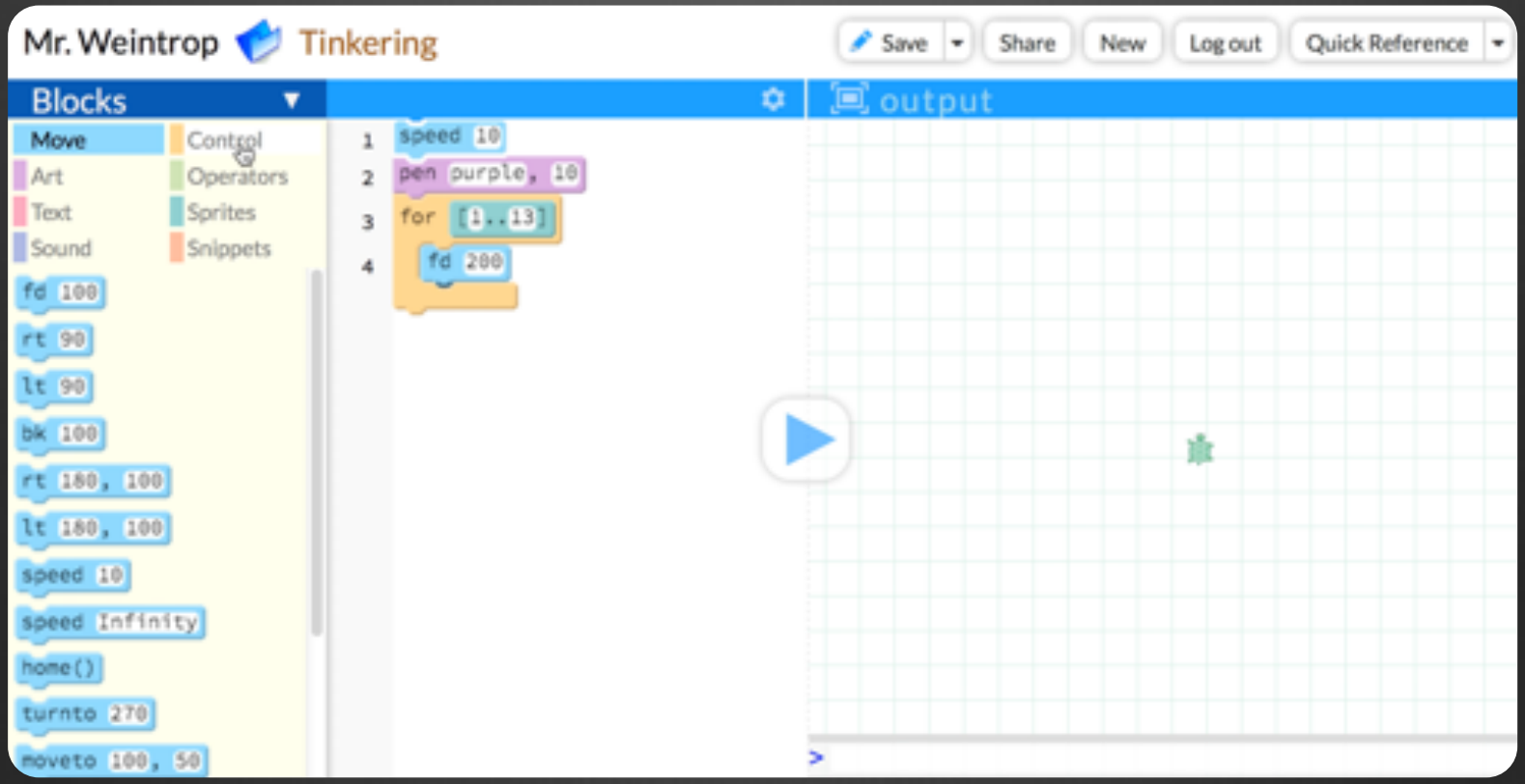



Hybrid Blocks/Text

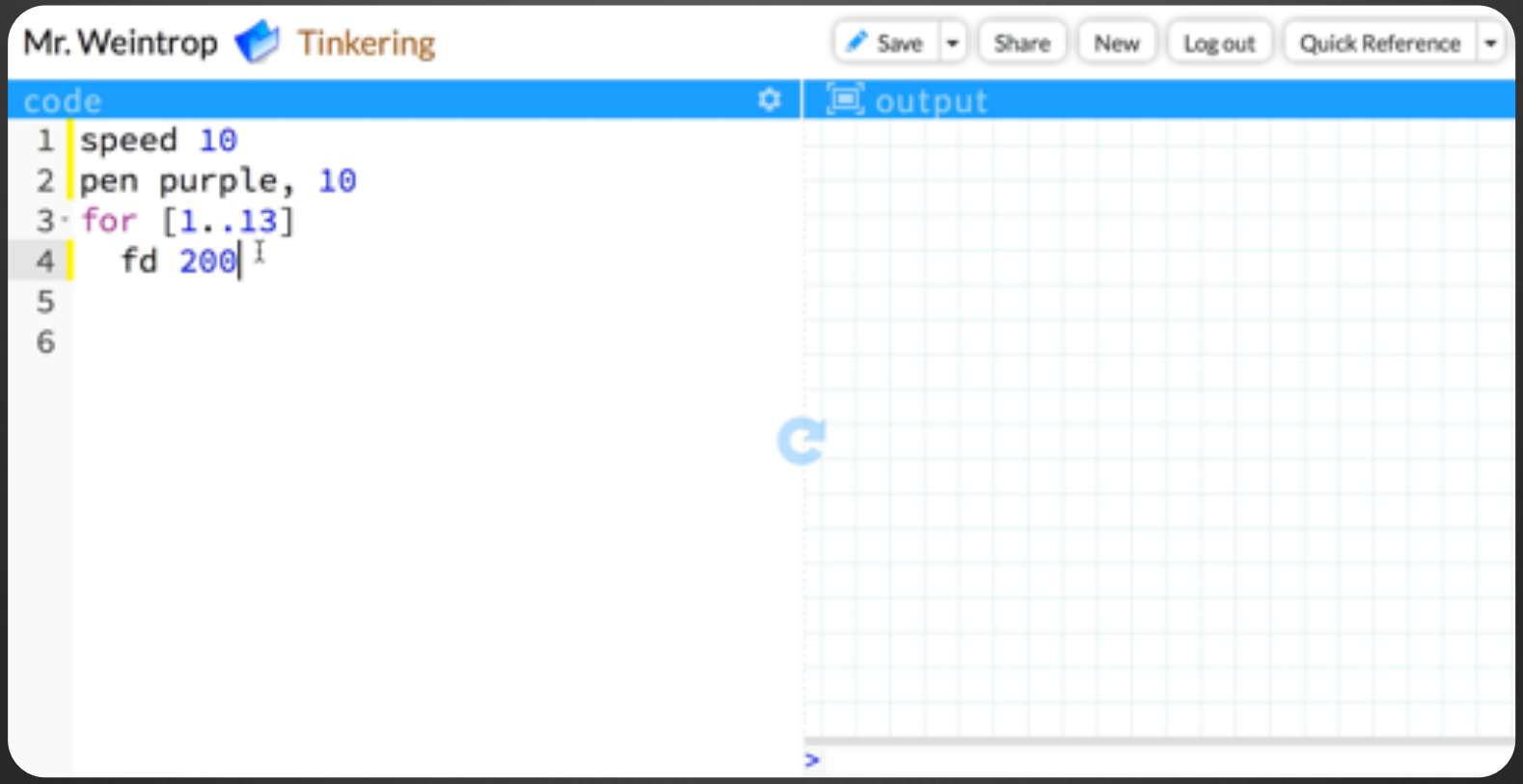



Text-based

# Block-based

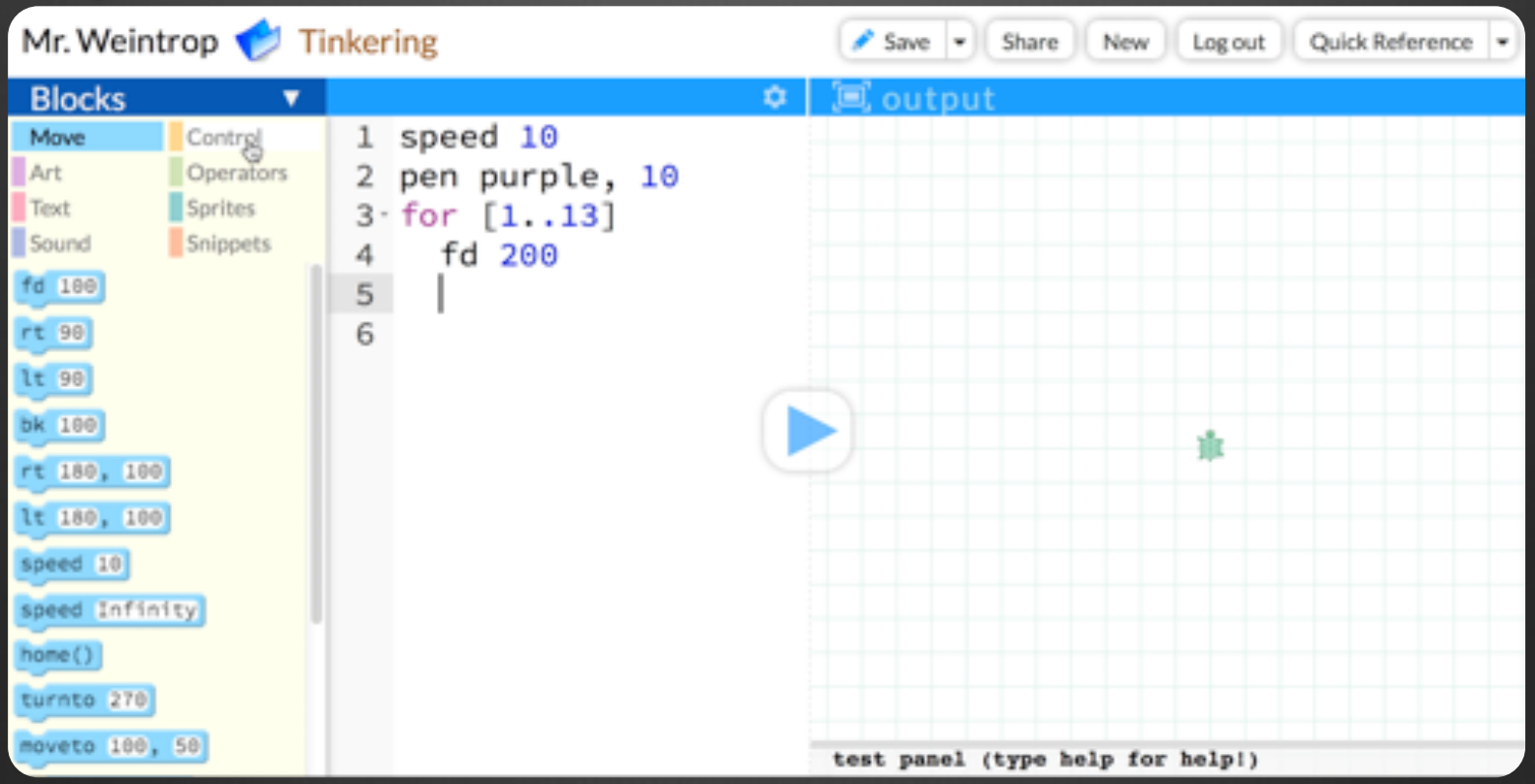

Mr. Weintrop
Tinkering
Save
Share
New
Log out
Quick Reference
Blocks
output
Move
Control
Art
Operators
Text
Sprites
Sound
Snippets
fd 100
rt 90
lt 90
bk 100
rt 180, 100
lt 180, 100
speed 10
speed Infinity
home()
turnto 270
moveto 100, 50
1 speed 10
2 pen purple, 10
3 for [1..13]
4 fd 200
>

# Text-based


Mr. Weintrop
Tinkering
Save
Share
New
Log out
Quick Reference
code
output
1 speed 10
2 pen purple, 10
3 for [1..13]
4 fd 200
5
6
>

# Hybrid

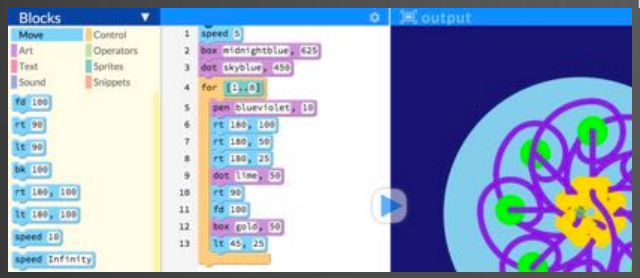

Block / Text


Mr. Weintrop
Tinkering
Save
Share
New
Log out
Quick Reference
Blocks
Move
Control
Art
Operators
Text
Sprites
Sound
Snippets
fd 100
rt 90
lt 90
bk 100
rt 180, 100
lt 180, 100
speed 10
speed Infinity
home()
turnto 270
moveto 100, 50
1 speed 10
2 pen purple, 10
3 for [1..13]
4 fd 200
5
6
output
test panel (type help for help!)

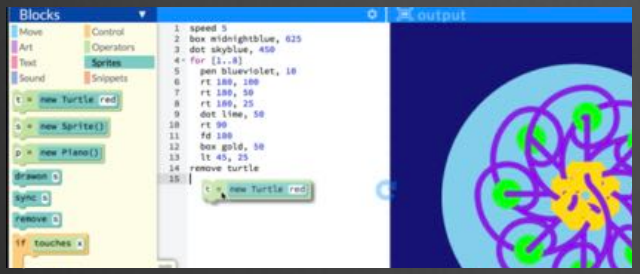
# Study Design

## 3 Conditions

Block-based

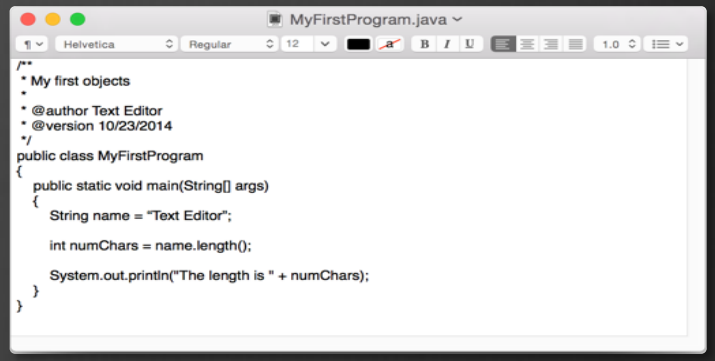



Hybrid Blocks/Tex




Text-based


Classroom-based, 3-condition, 2-phase, Quasi-Experimental, Mixed Method Study




First day of School

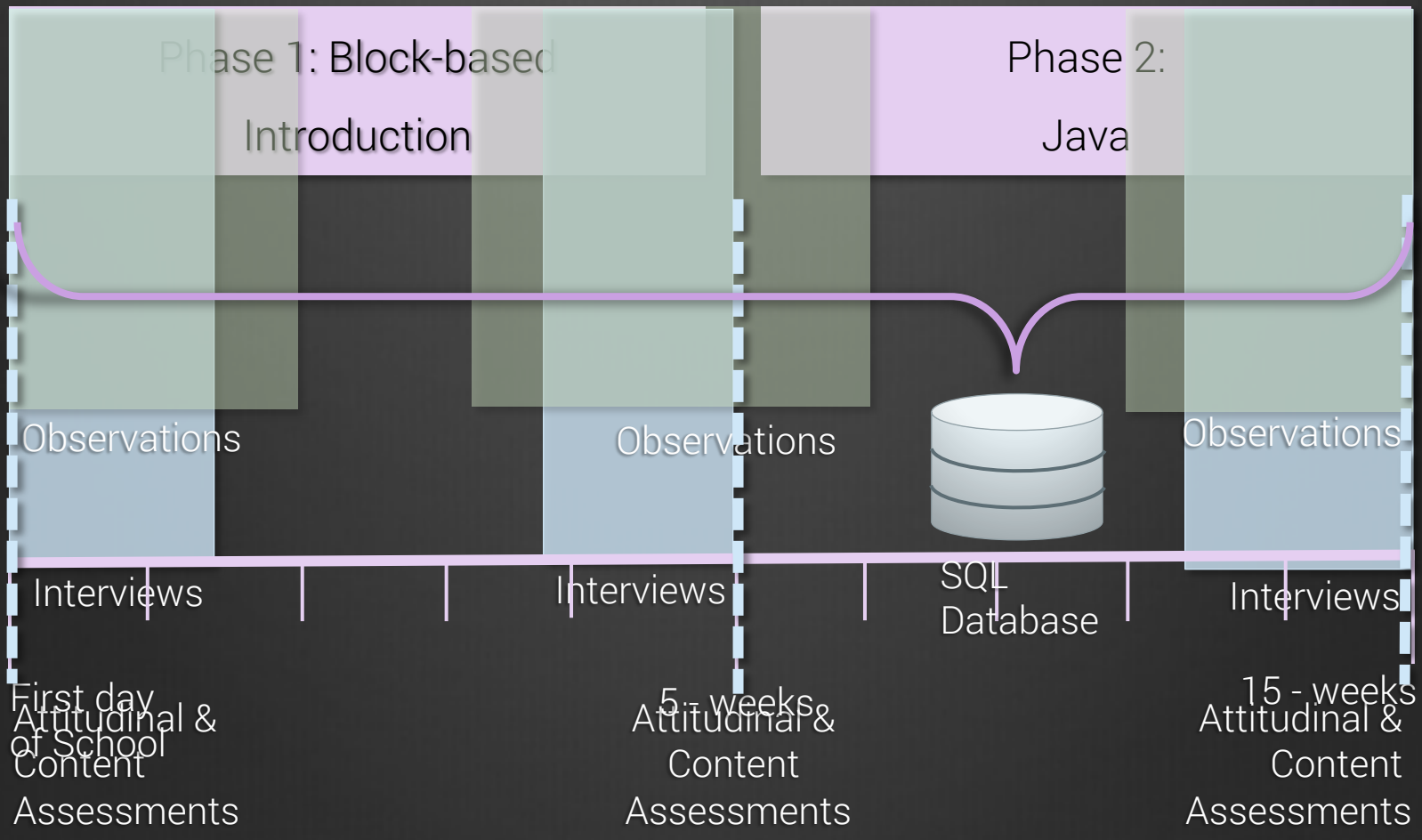
Phase 1: Block-based Introduction

5 - weeks

Phase 2: Java

15 - weeks

# Data Collection Schedule



# Data Collected

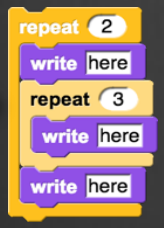
## Quantitative

**Content Survey:**
8500+ Responses
74 Pre/Mid/Post sets

**Attitudinal Survey:**
81 Pre/Mid/Post sets

How many times will the word “here” be printed?

```
repeat 2
  write "here"
  repeat 3
    write "here"
  write "here"
```



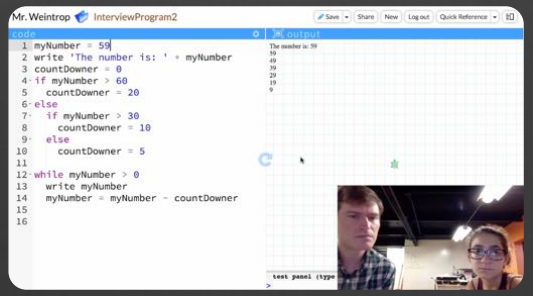

I will do well in this class.
I think programming is fun.
I am excited for this class.

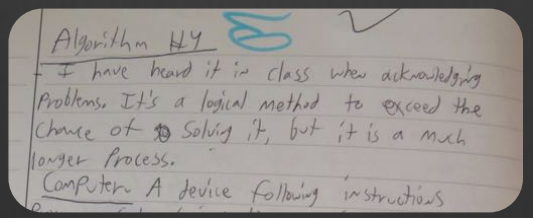
## Qualitative

**Interviews**
12 Pre; 10 Mid; 13 Post

**Observations**
40 classroom observations

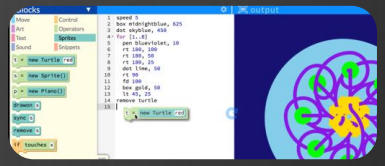
**Artifacts**
87 Student Journals
88 Projects/Presentations

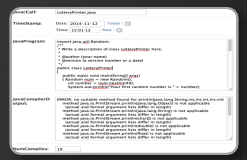


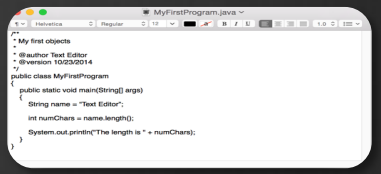




## Computational

**147,000+ Pencil.cc Events**
**11,000+ Java Runs**
**6,000+ Java Errors**





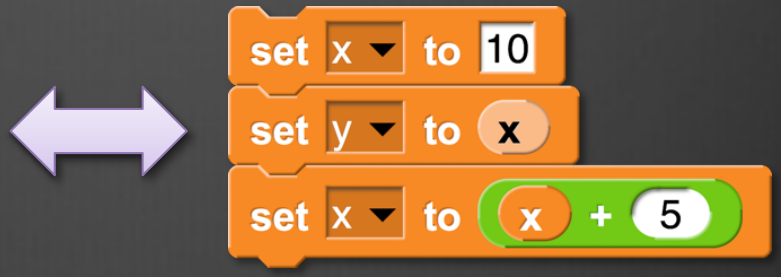


*How do you Measure Content Understanding By Modality?*

# The Commutative Assessment

What will be the value of x and y after this script is run?

```
var x = 10;
var y = x;
x = (x + 5);
```




A) x is equal to 15 and y is equal to 15

B) x is equal to 5 and y is equal to 10

C) x is equal to 15 and y is equal to 10

D) x is equal to “x + 5” and y is equal to “x”

E) x is equal to 10, 15 and y is equal to 10

# The Commutative Assessment

Variable values become linked (Du Boulay, 1986)

Variable values are singletons that get passed on assignment (Du Boulay, 1986)

A) **x** is equal to 15 and **y** is equal to 15

B) **x** is equal to 5 and **y** is equal to 10

Correct Answer

C) **x** is equal to 15 and **y** is equal to 10

D) **x** is equal to "x + 5" and **y** is equal to "x"

Expressions "do not evaluate" (Bayman & Mayer 1983; Sorva, 2008)

E) **x** is equal to 10, 15 and **y** is equal to 10

Variables remember prior values (Du Boulay1986; Doukakis et al., 2007)

# *Findings*

# Perceptions of  Programming


| Perception | Frequency |
|---|---|
| Blocks-based Programming is Easier | 88% |
| Text-based Programming is Easier | 8% |
| Comparable Difficulty | 4% |

*"Learning Java is more complicated than [Pencil.cc]"*

"[In Java] *there are no blocks to help out, it is basically done from scratch"*

# Perceptions of Block-based Programming

## Perceived Affordances

*Java is not in English it's in Java language, the blocks are in English, it's easier to understand.*

**Blocks are Easier to Read**

*If [the block is] rounded or diagonal, they'll know the difference...It's like a puzzle.*

**Shape and Visual Layout**

*It's just harder to write with the codes*

**Easier to Compose**

*[The blocks] kind of jog your memory*

**Browsability**

*In Java, I will not be able to test out blocks and incorporate them and see if they work.*

**Support for Trial & Error**

*There will be no set blocks that will provide you with pre-made functions*

**Prefabricated Commands**

*Snap! was more about making sprites do things*

**Visual Outcomes**

# Perceptions of Block-based Programming

## Perceived Drawbacks

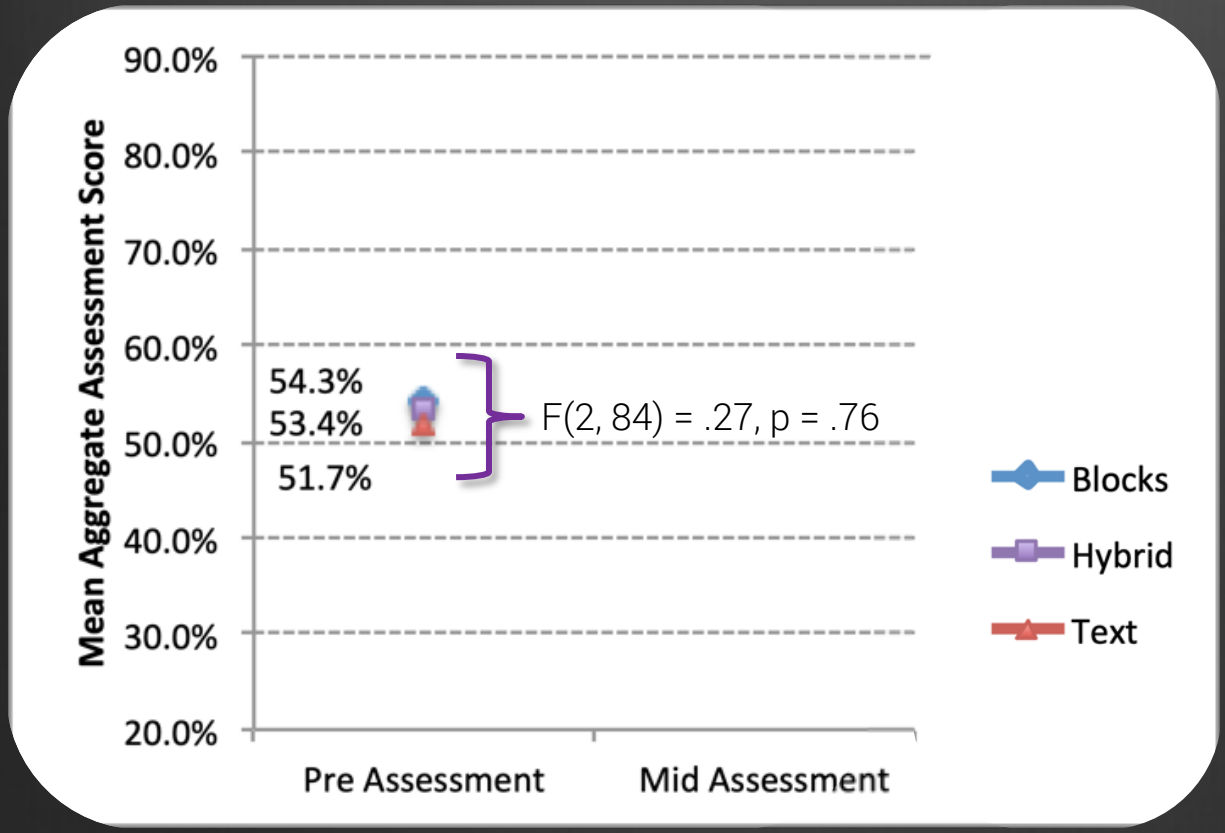
*Blocks are limiting…There is not a block for everything.*

*It seems like when there is more blocks it's more confusing*

*If we actually want to program something, we wouldn't have blocks*

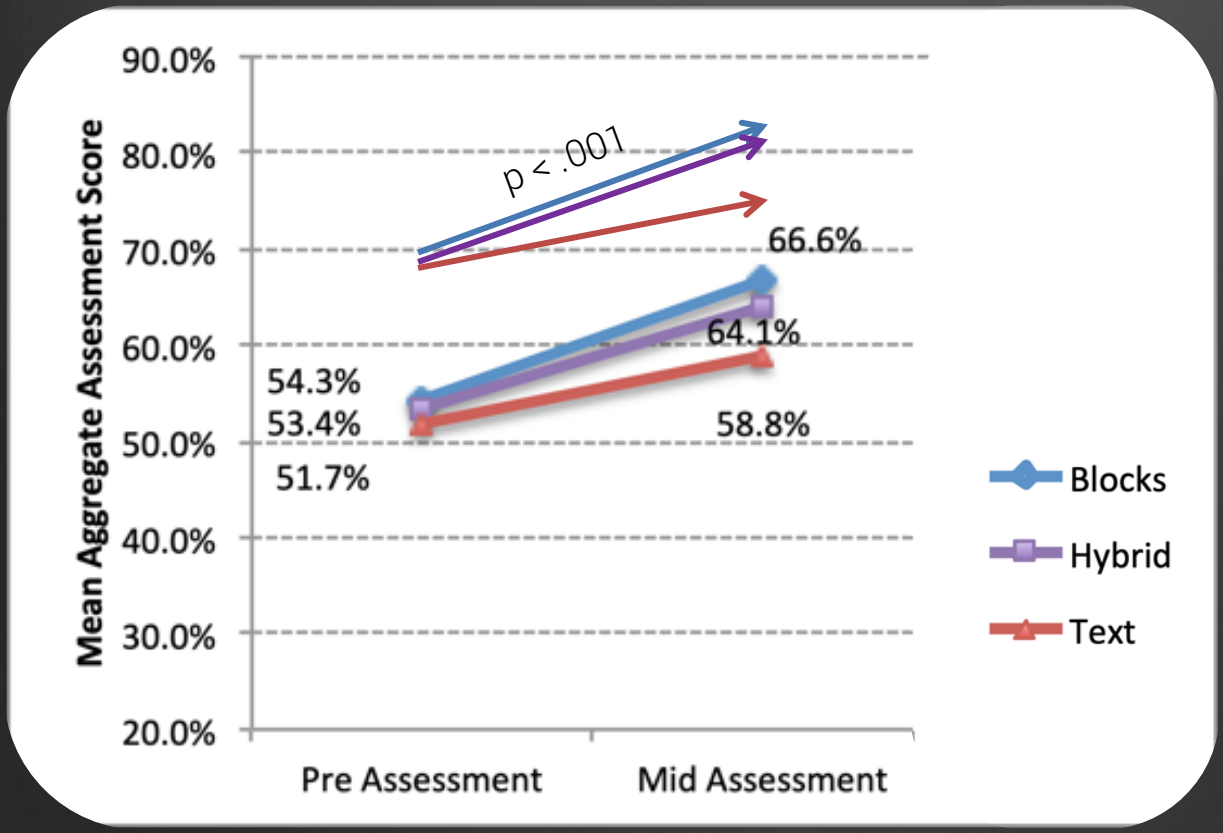

[IDC - Weintrop & Wilensky, 2015]

# Learning Outcome by Condition



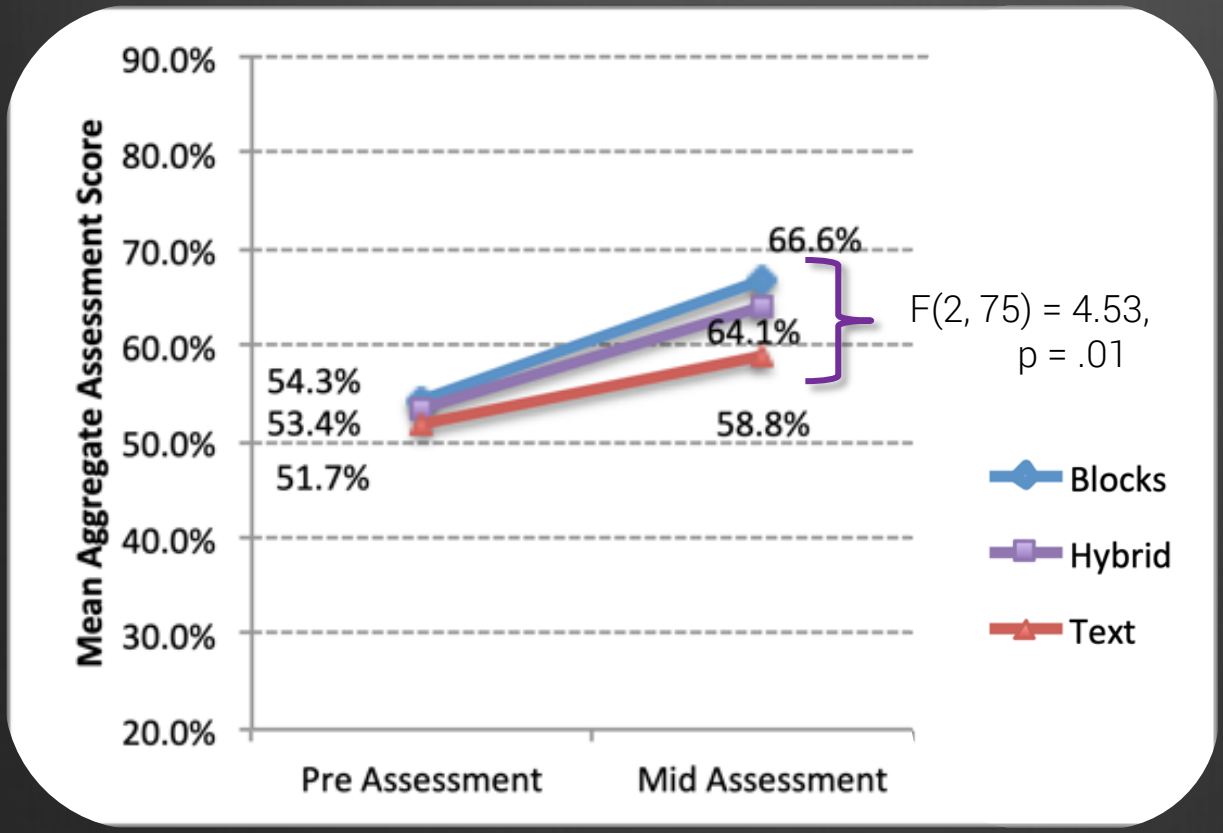

[TOCE - Weintrop & Wilensky, 2017]

# Learning Outcome by Condition



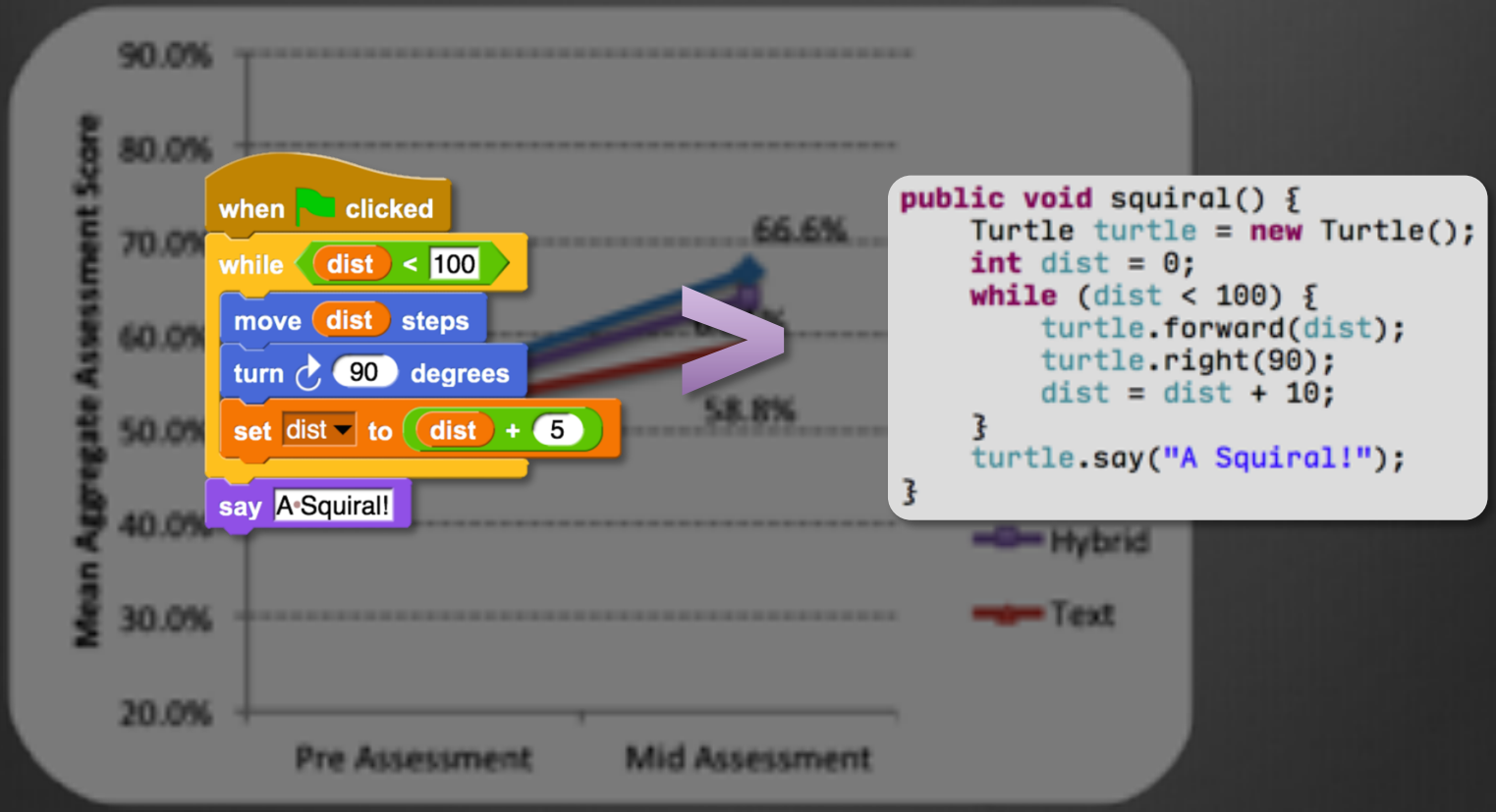

[TOCE - Weintrop & Wilensky, 2017]

# Learning Outcome by Condition



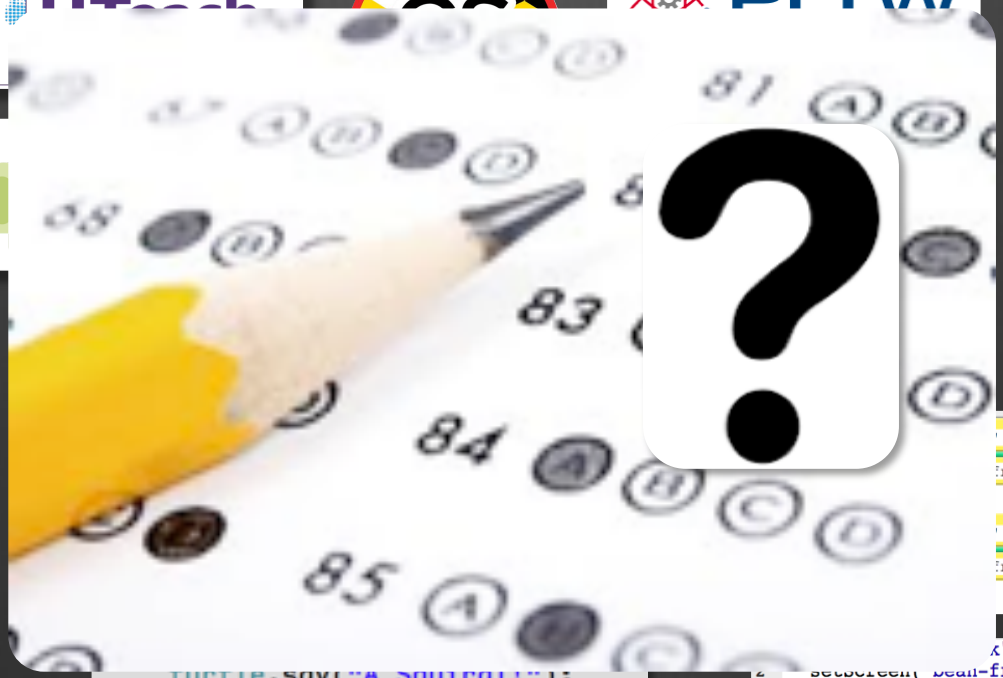

[TOCE - Weintrop & Wilensky, 2017]

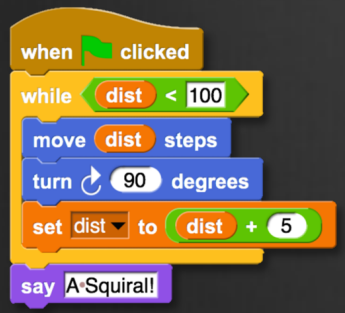
# Learning Outcome by Condition

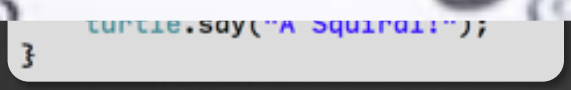


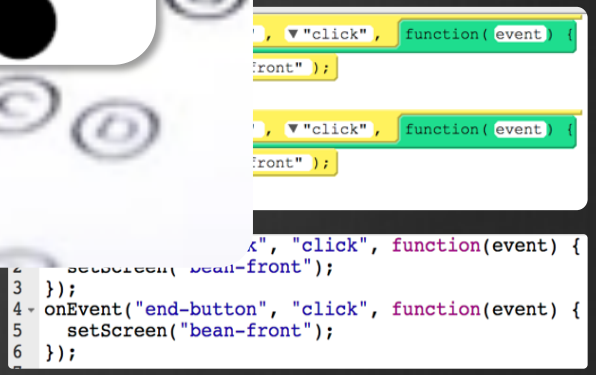

[TOCE - Weintrop & Wilensky, 2017]

# AP Computer Science Principles






## Assignment

```
a ← expression
```


## Conditional Logic

```
IF (condition)
{
   <block of statements>
}
```


## Iterative Logic

```
REPEAT n TIMES
{
   <block of statements>
}
```


## Output

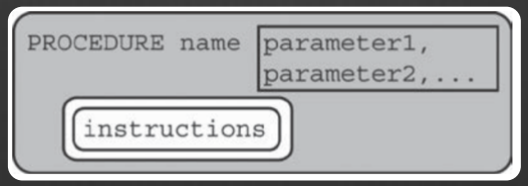
```
DISPLAY (expression)
```


## Robot Commands

```
MOVE_FORWARD ()
```

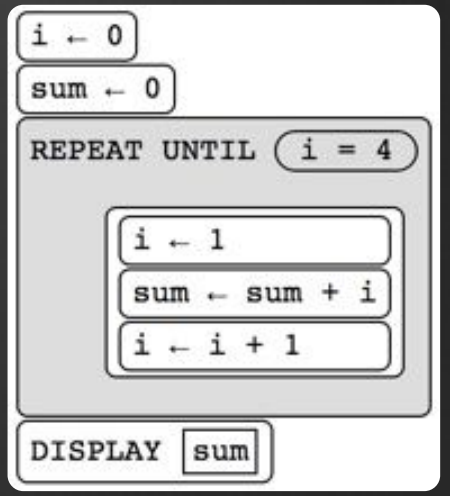

## Procedures

```
PROCEDURE name (parameter1,
                parameter2, ...)
{
     <instructions>
}
```

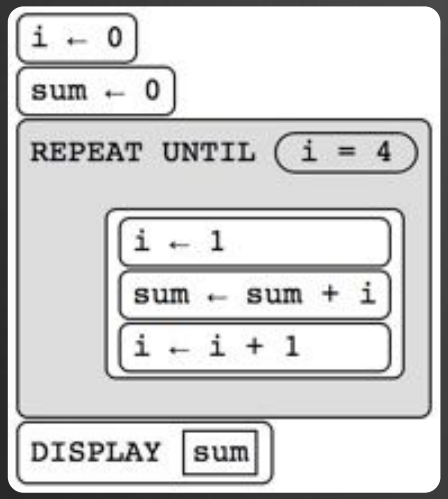



## Programs

```
i ← 0
sum ← 0
REPEAT UNTIL (i = 4)
{
   i ← 1
   sum ← sum + i
   i ← i + 1
}
DISPLAY (sum)
```

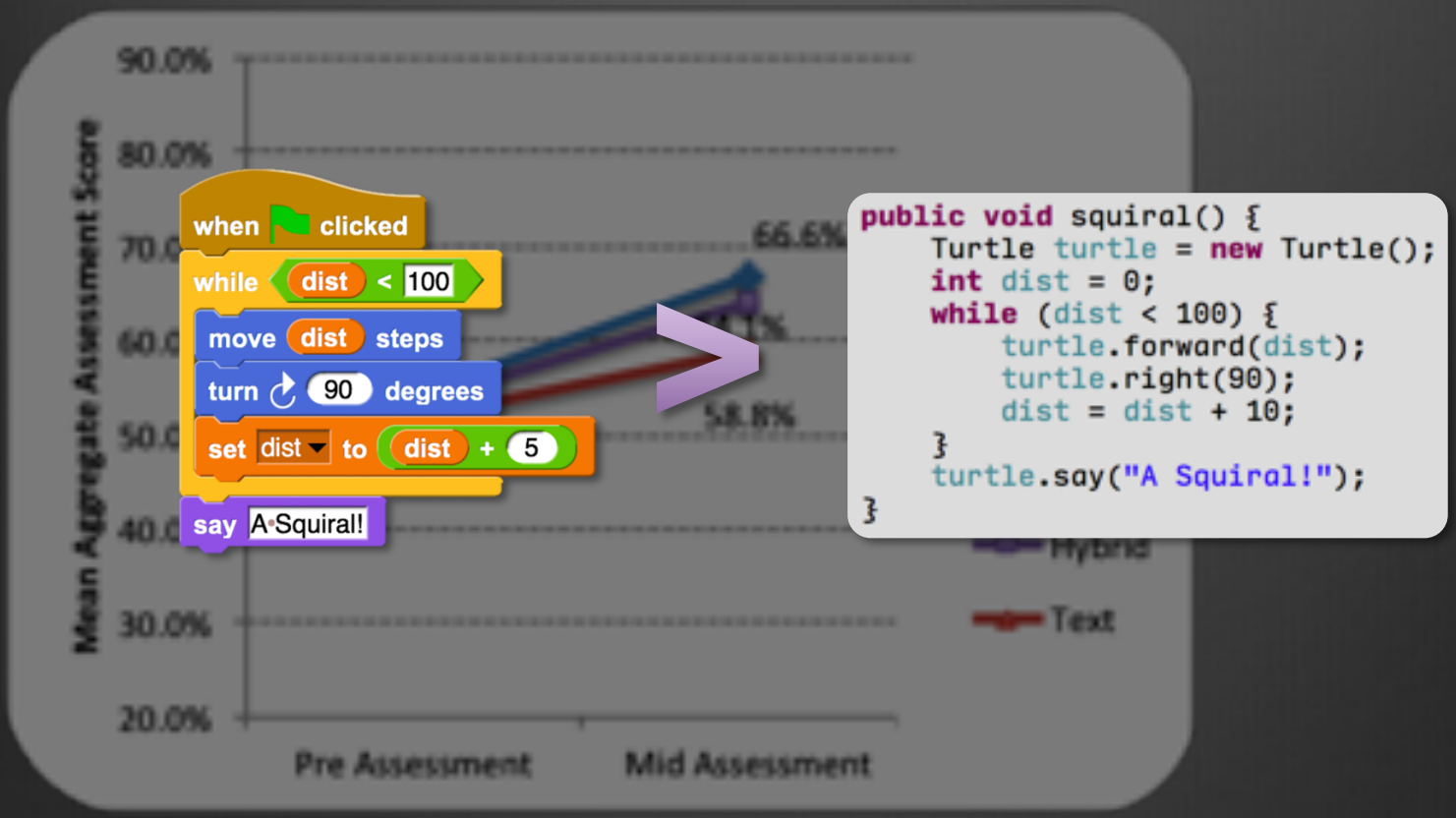



# Perceived Affordances



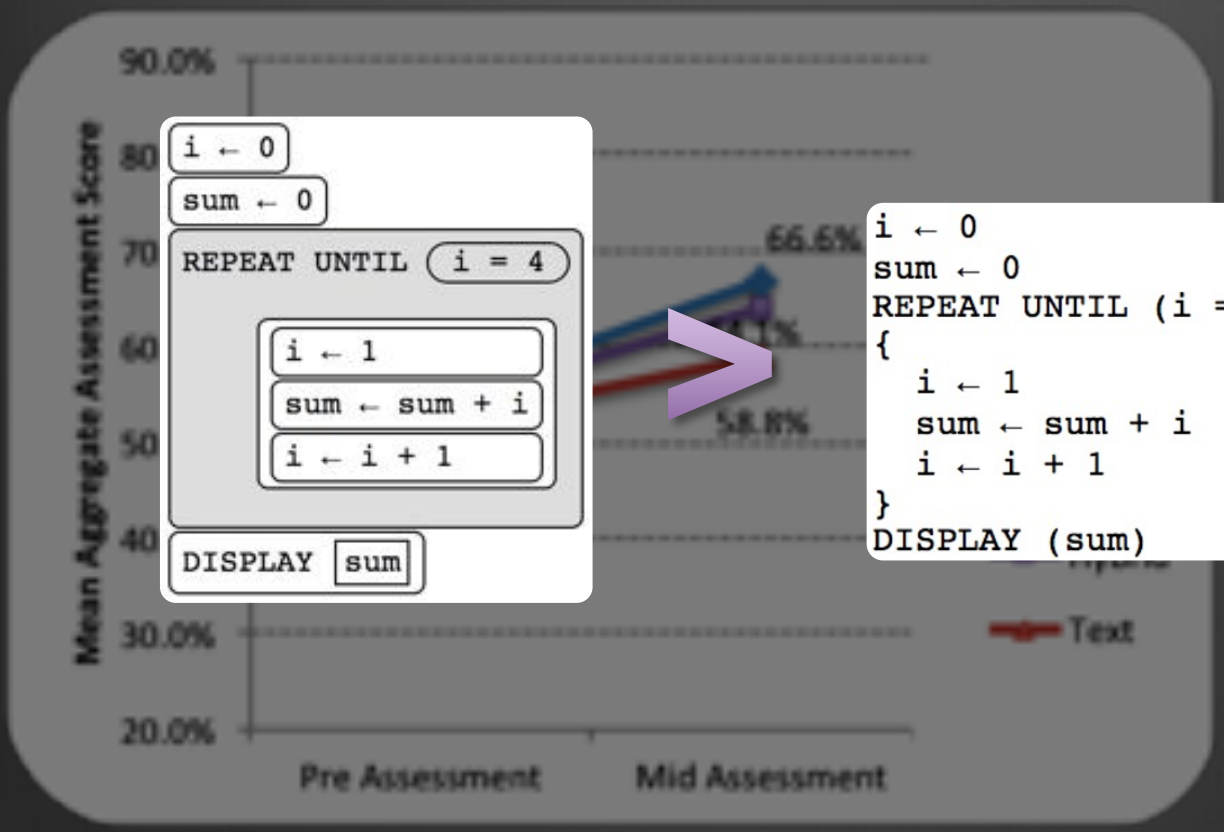

- Blocks are Easier to Read
- Shape and Visual Layout
- Easier to Compose
- Browsability
- Support for Trial & Error
- Prefabricated Commands
- Visual Outcomes

# AP Computer Science Principles




[TOCE - Weintrop & Wilensky, 2017]

# AP Computer Science Principles




```
i ← 0
sum ← 0
REPEAT UNTIL ( i = 4 )
    i ← 1
    sum ← sum + i
    i ← i + 1
DISPLAY [ sum ]
```

>

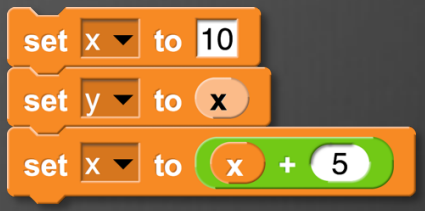
```
i ← 0
sum ← 0
REPEAT UNTIL (i = 4)
{
   i ← 1
   sum ← sum + i
   i ← i + 1
}
DISPLAY (sum)
```

5) = 4.53,
p = .01

[TOCE - Weintrop & Wilensky, 2017]

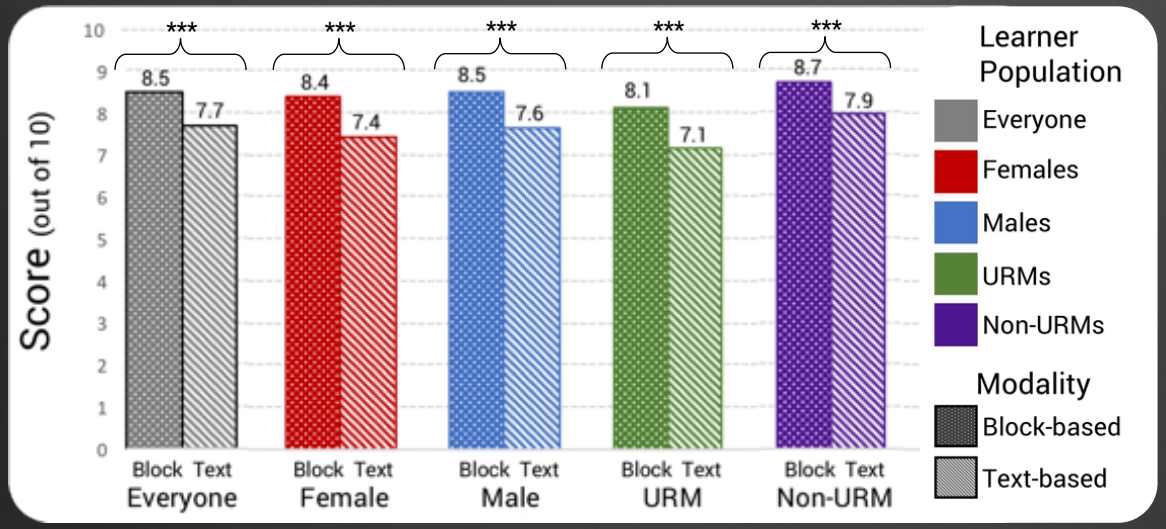
# AP Computer Science Principles

```
var x = 10;
var y = x;
x = (x + 5);
```



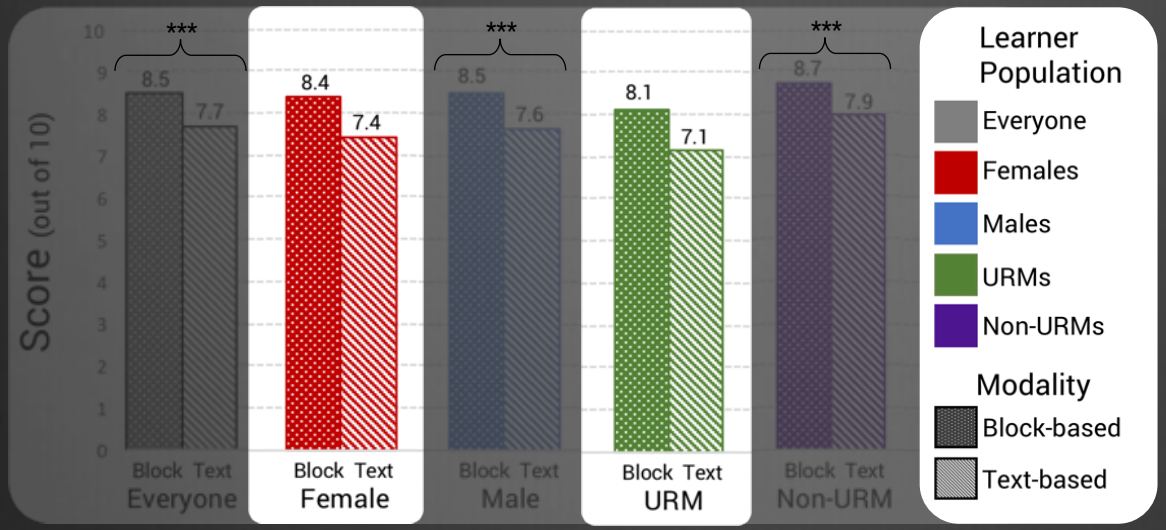

```
x ← 10
y ← x
x ← x + 5
```

```
x ← 10
y ← x
x ← x + 5
```


All students perform better on block-based questions.

[ICLS - Weintrop et al., 2018; SIGCSE - Weintrop et al., 2019]

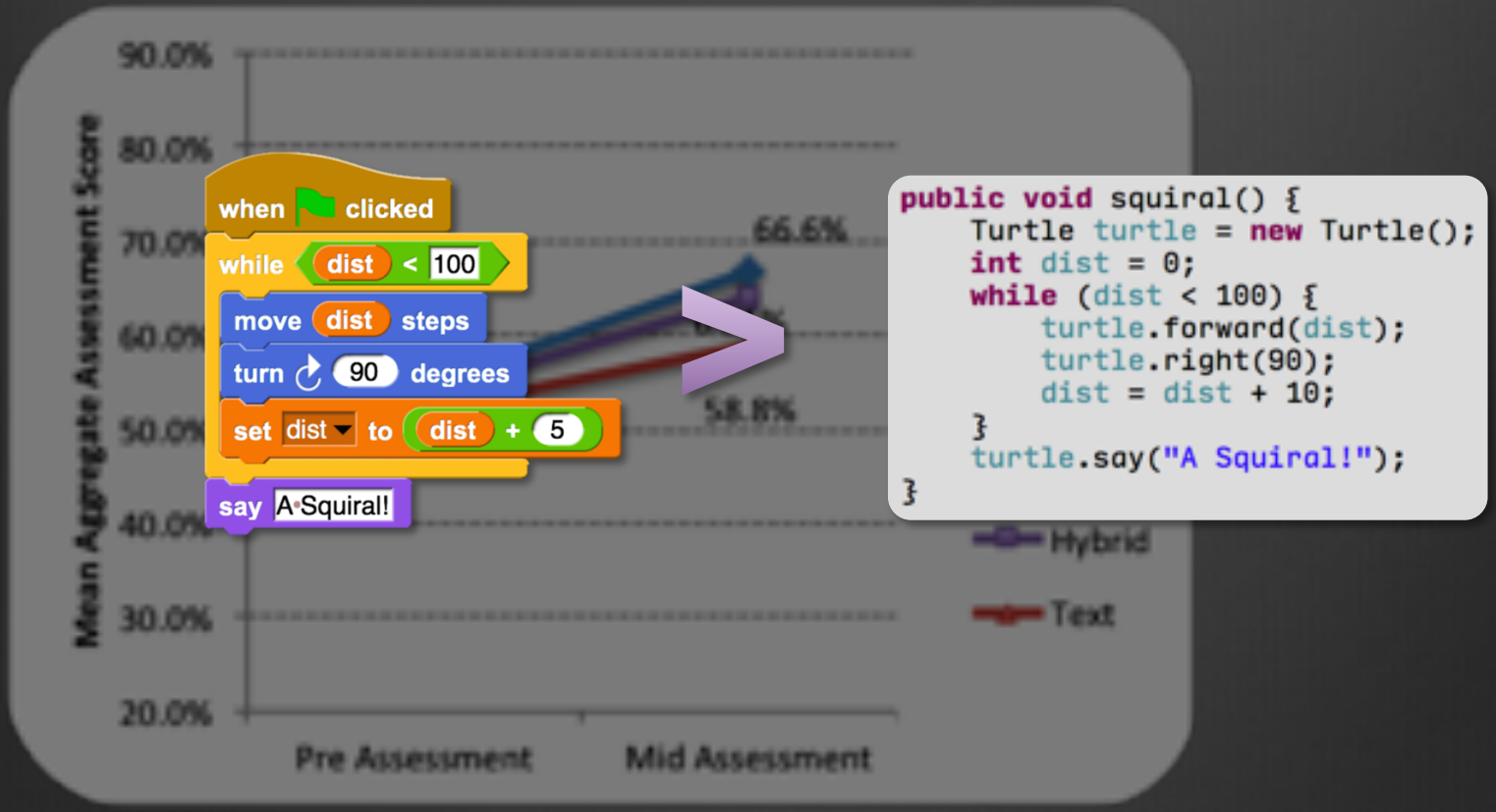




All students perform better on block-based questions.

Students from historically underrepresented populations see a greater benefit in the shift from text-based to block-based programming.

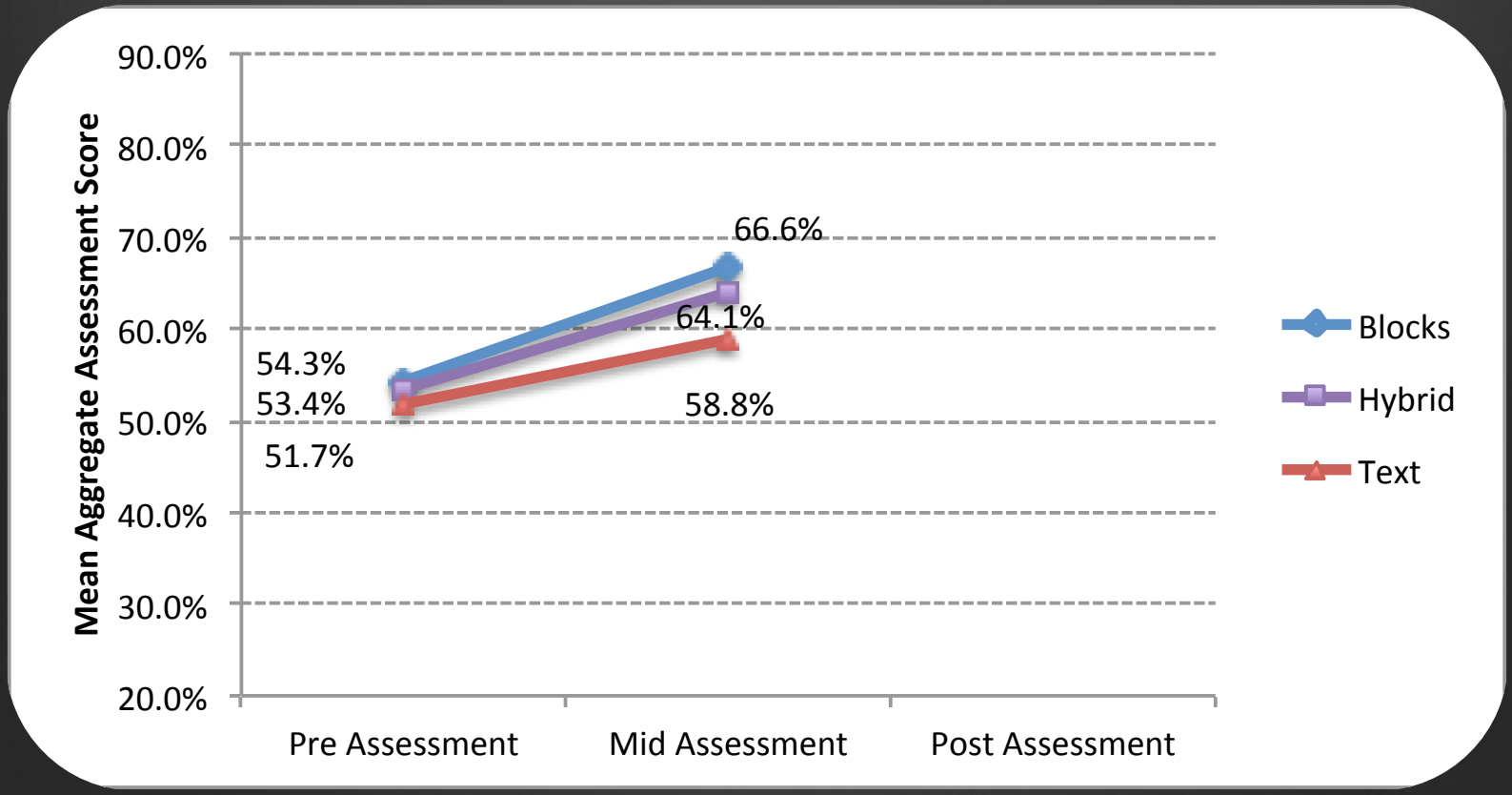
[ICLS - Weintrop et al., 2018; SIGCSE - Weintrop et al., 2019]

# Learning Outcome by Condition



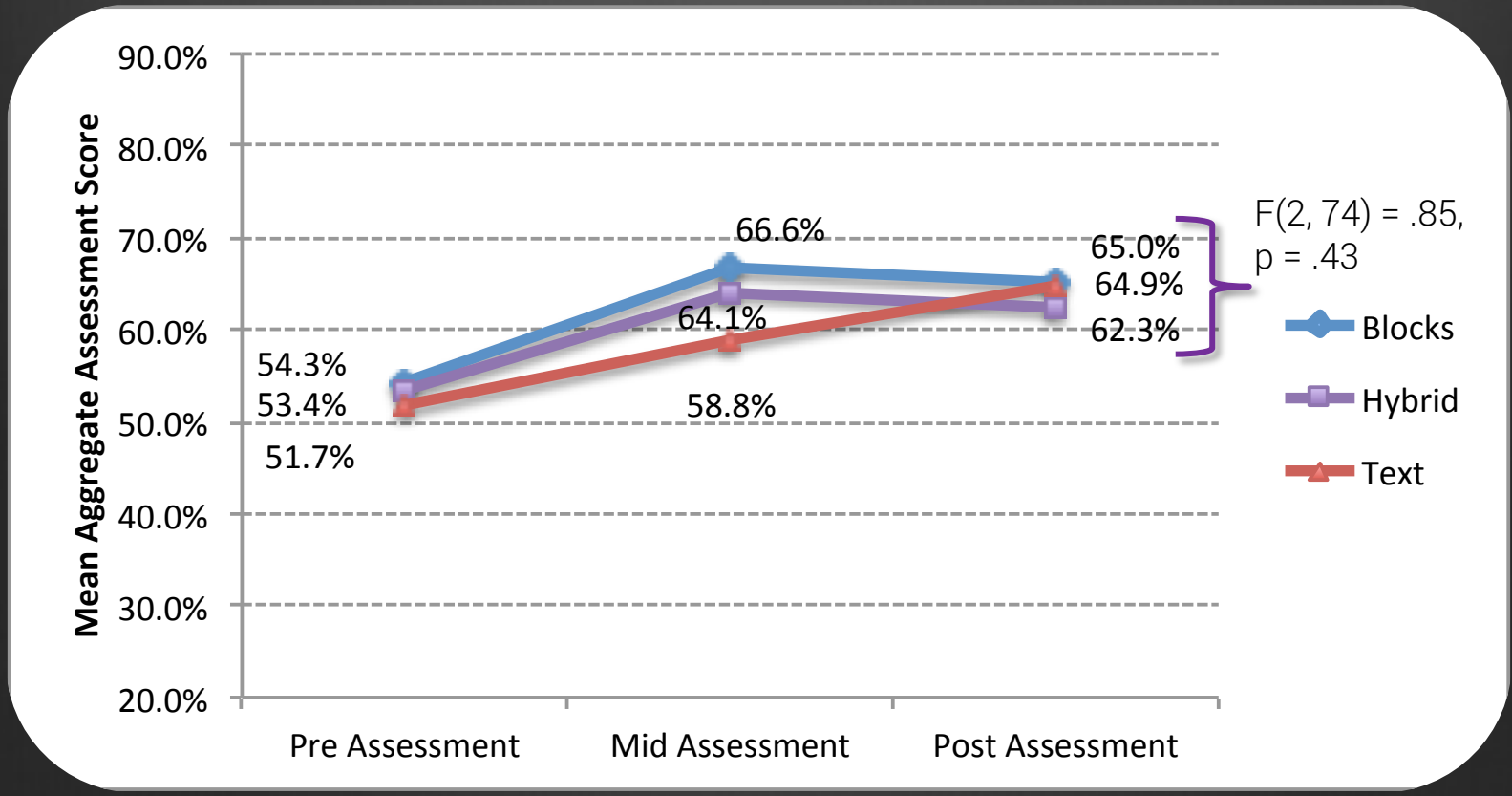

[TOCE - Weintrop & Wilensky, 2017]

# Learning Outcome by Condition




[Computers & Education - Weintrop & Wilensky,2019]

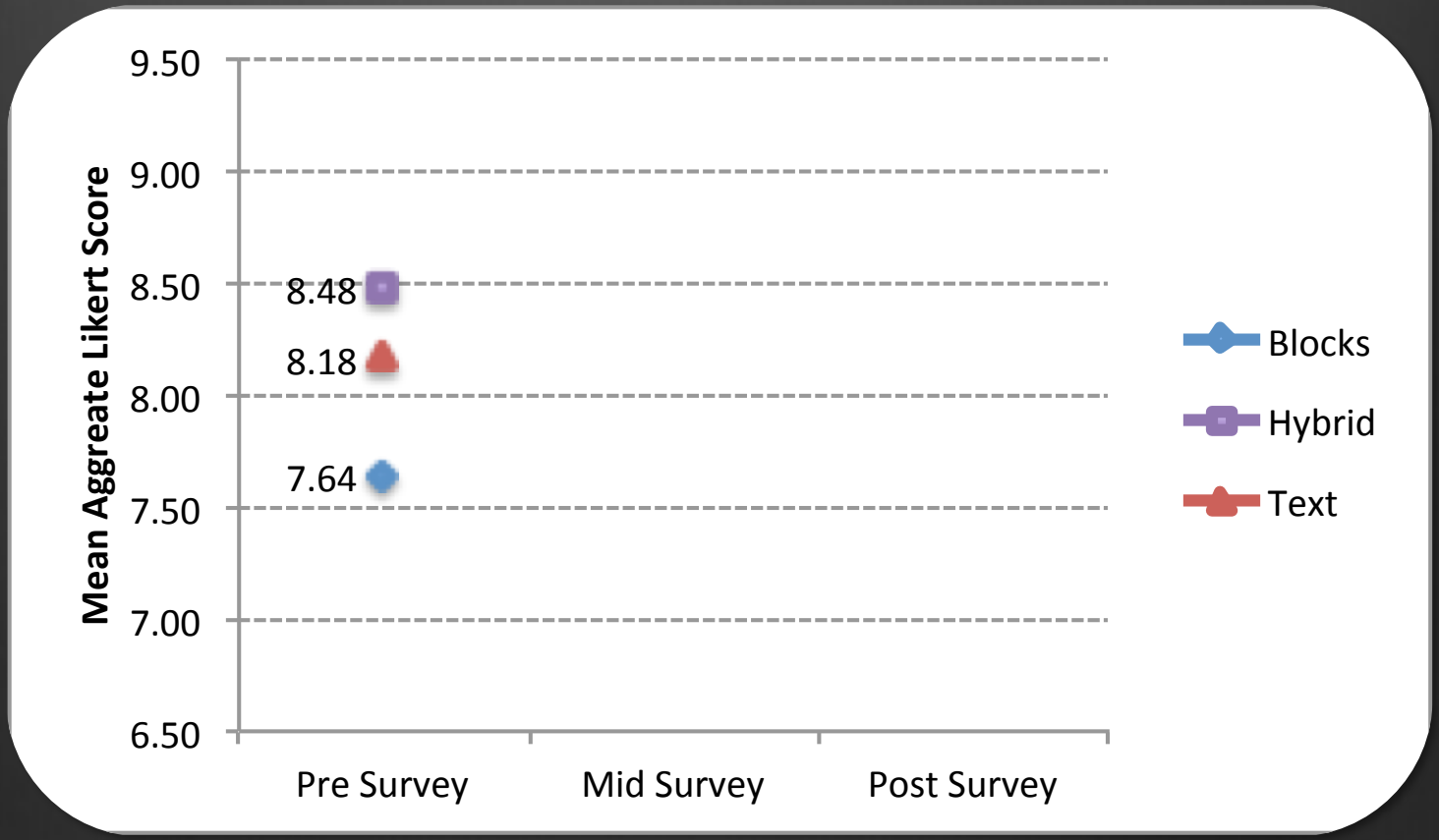
# Learning Outcome by Condition



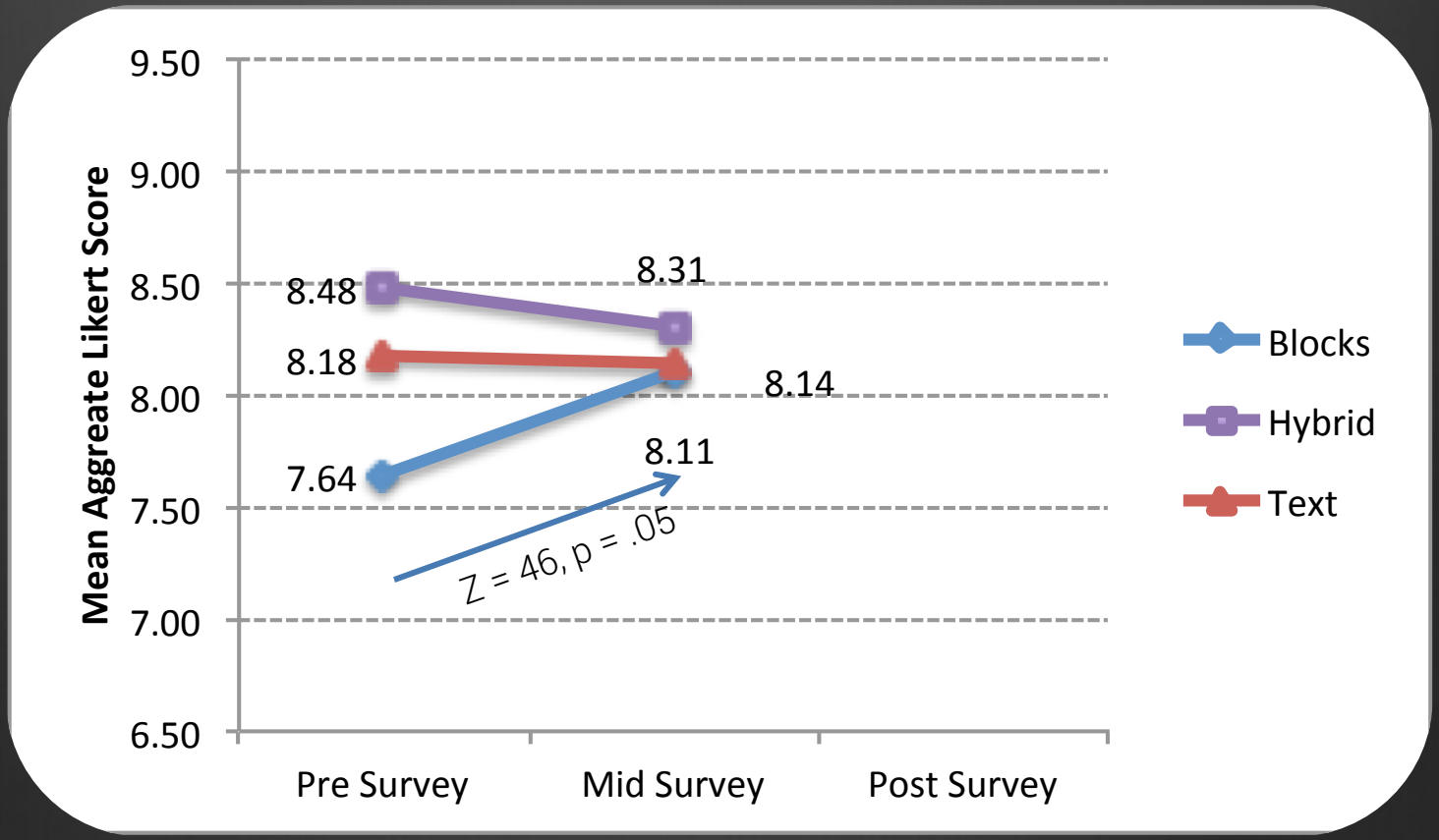

[Computers & Education - Weintrop & Wilensky,2019]

# *Perceptual Findings & Attitudinal Outcomes By Condition*

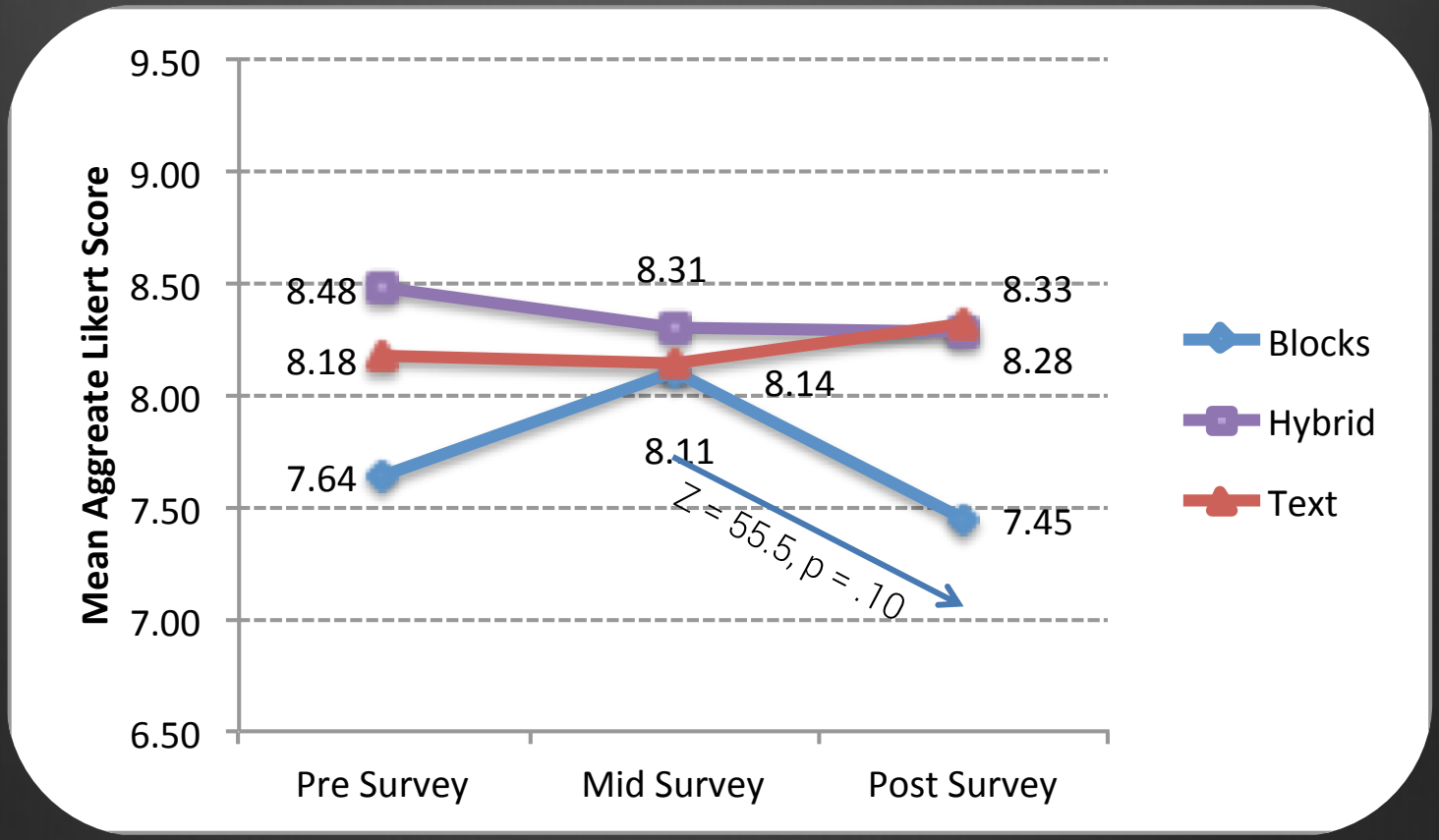
# Aggregate Confidence Score

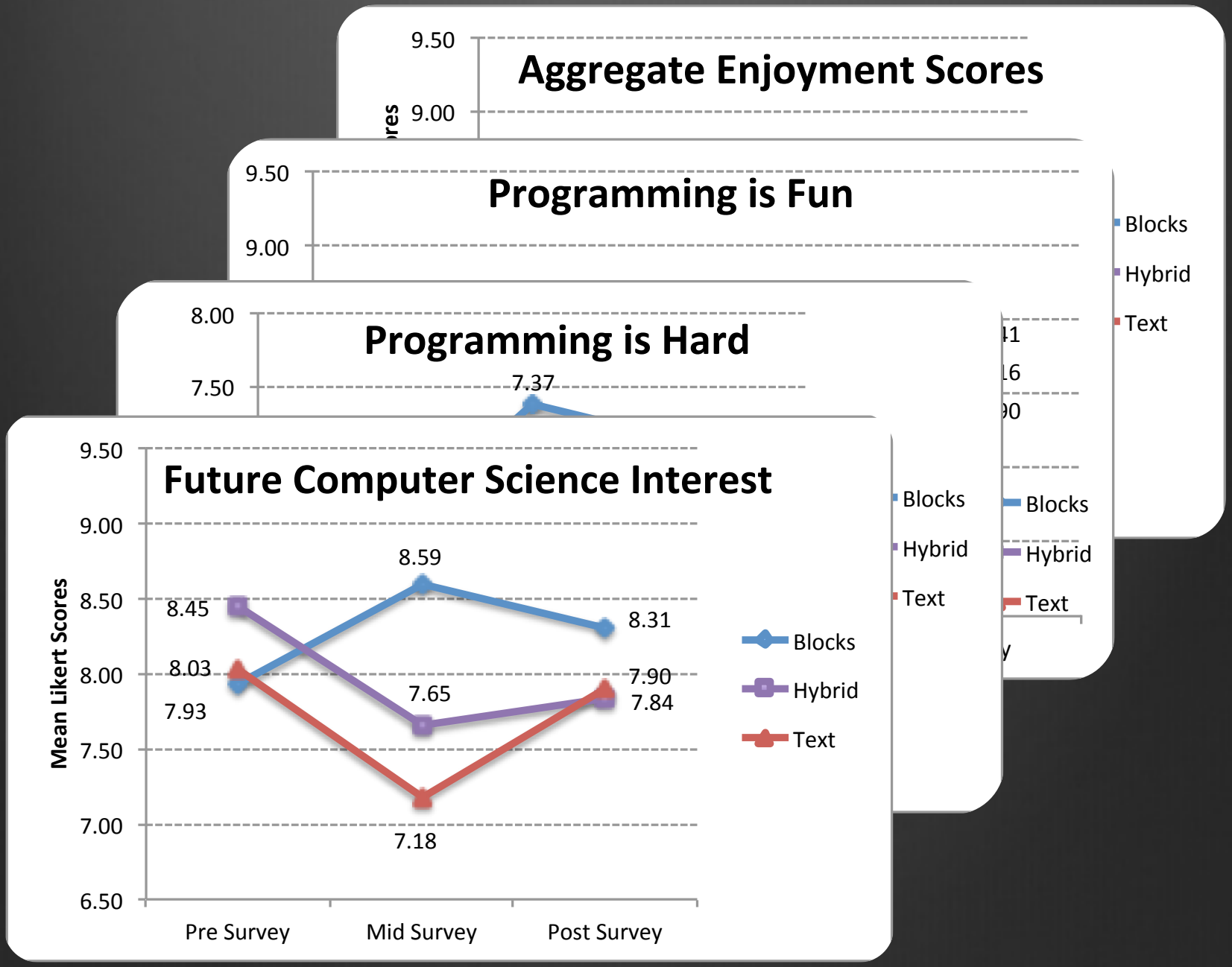


# Aggregate Confidence Score

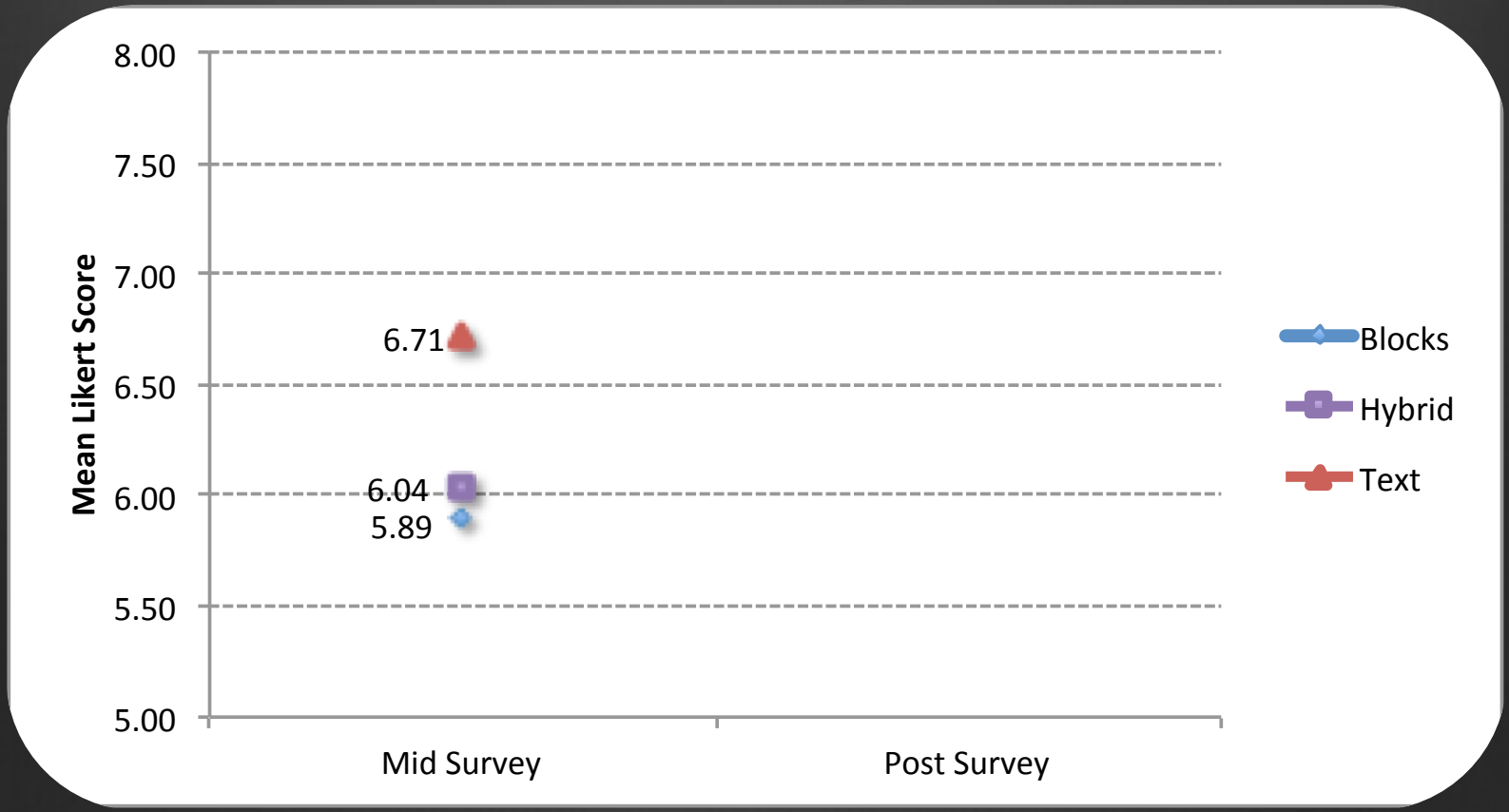


# Aggregate Confidence Score



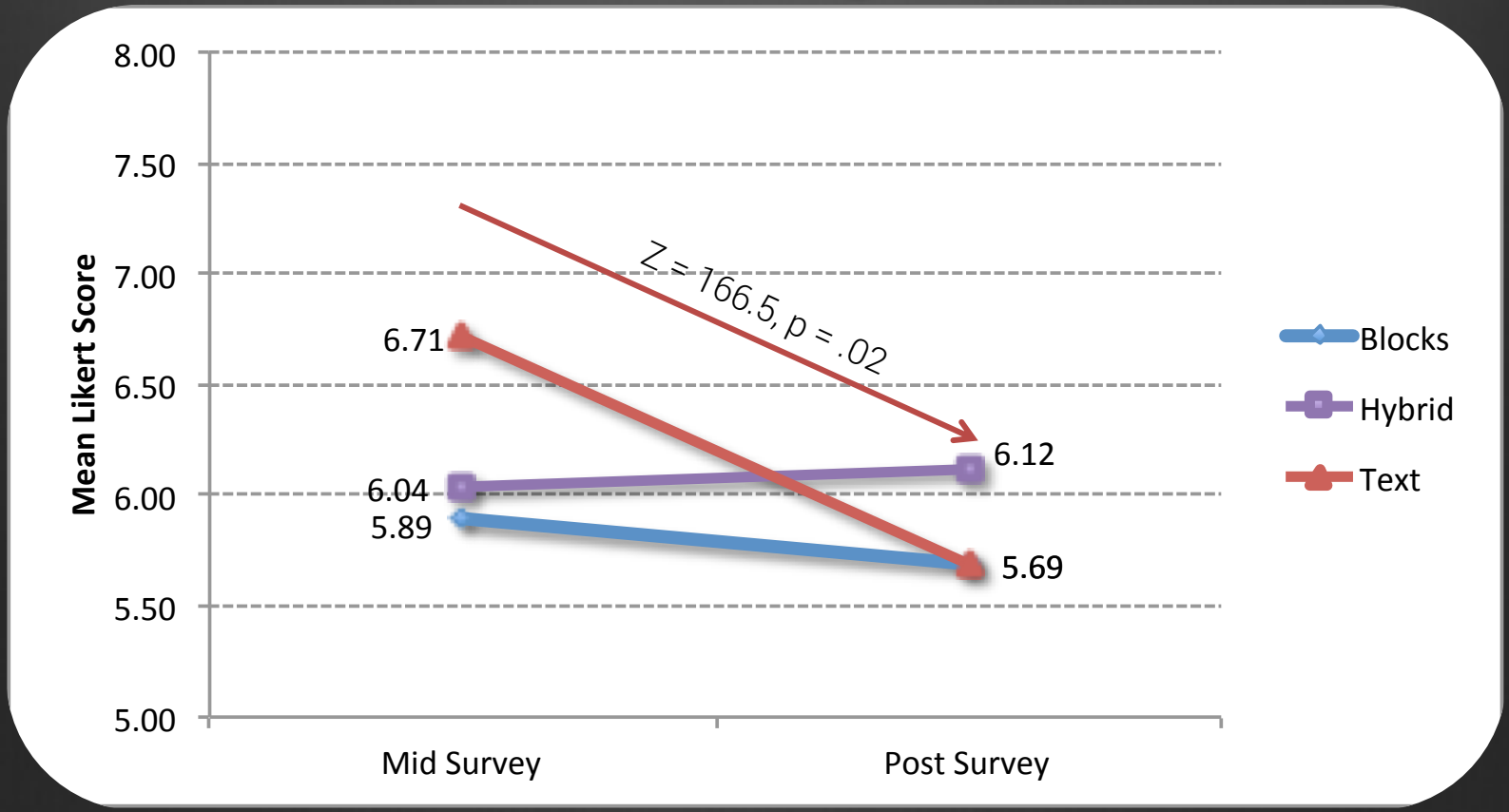

Aggregate Enjoyment Scores
9.50
9.00
Programming is Fun
9.50
9.00
Blocks
Hybrid
Text
Programming is Hard
8.00
7.50
7.37
Future Computer Science Interest
9.50
9.00
8.50
8.00
7.50
7.00
6.50
Mean Likert Scores
8.59
8.45
8.31
8.03
7.65
7.90
7.84
7.93
7.18
Pre Survey
Mid Survey
Post Survey
Blocks
Hybrid
Text

# Pencil.cc is Similar to What Real Programmers Do




Year Two Findings

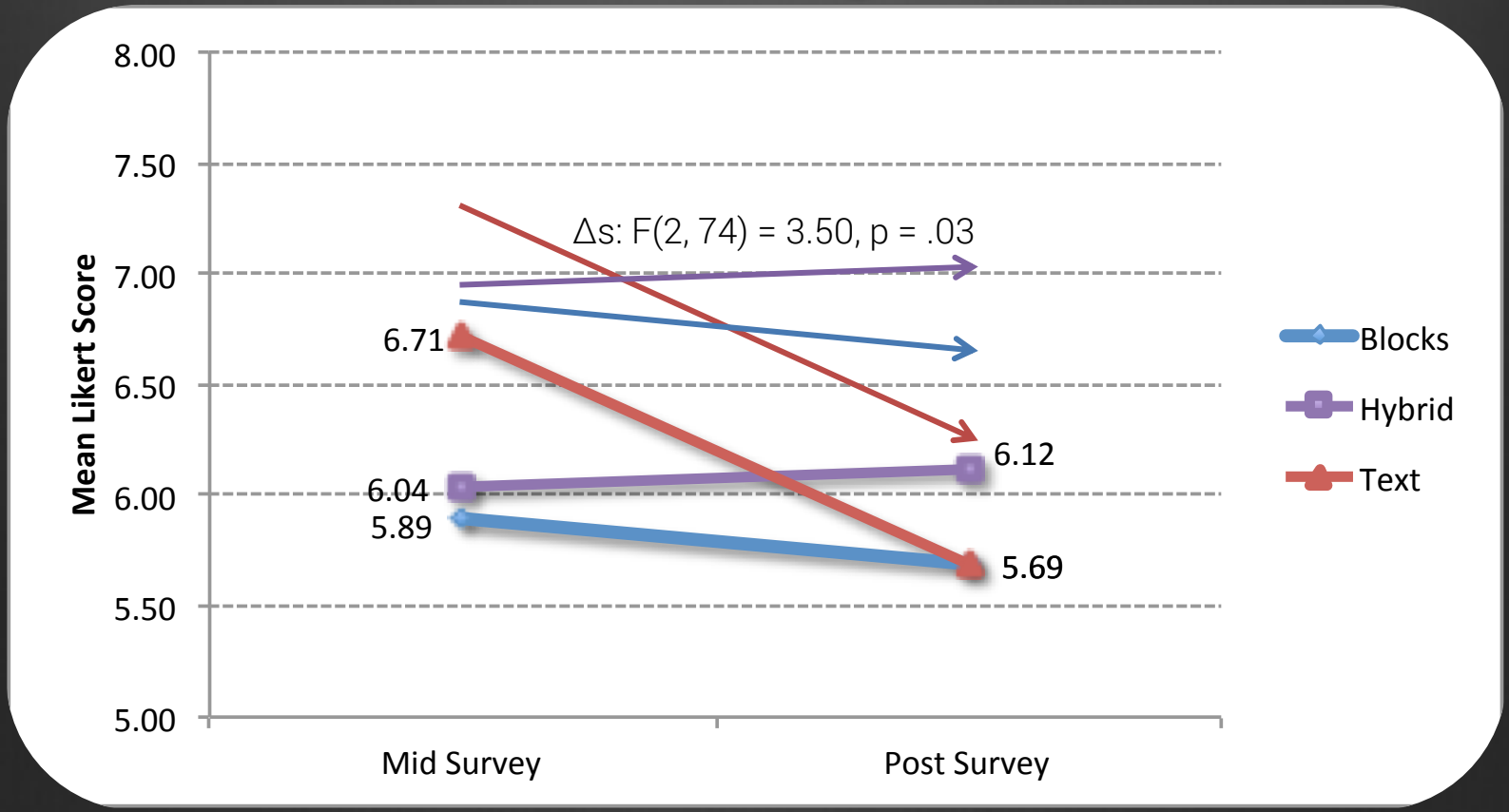
# Pencil.cc is Similar to What Real Programmers Do




*"It could form the basis of programming, but it's just basic stuff, not like professional or anything"*

Year Two Findings

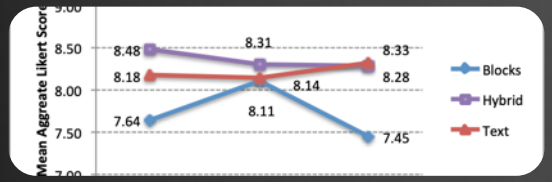
# Pencil.cc is Similar to What Real Programmers Do

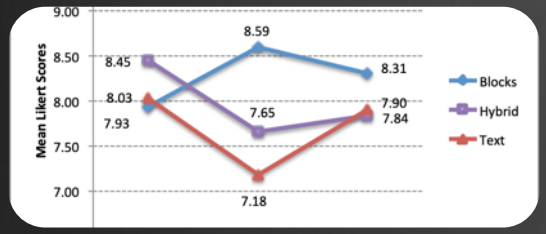


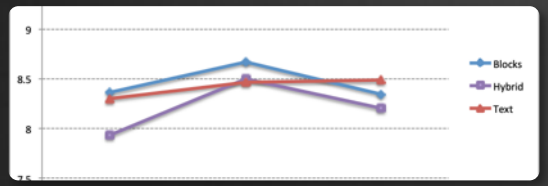

*"It could form the basis of programming, but it's just basic stuff, not like professional or anything"*

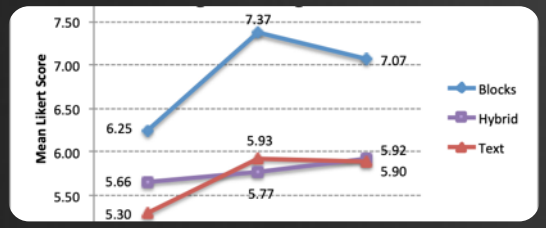
Year Two Findings

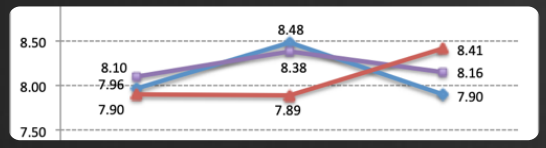
# Perceptual & Attitudinal Findings Discussion



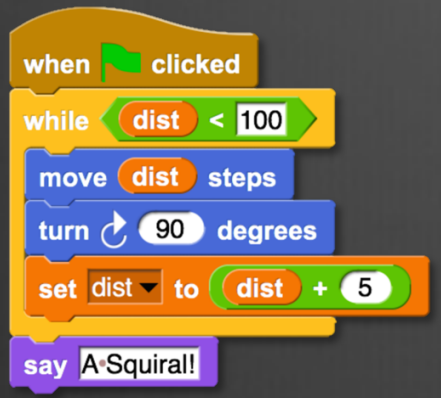

## General Trends:

Blocks condition: Positive -> Negative

Text condition: Negative/Flat -> Positive

Hybrid: A bit of Both, was seen as most helpful and most authentic

# *So, What Should I Use in my Classroom?*

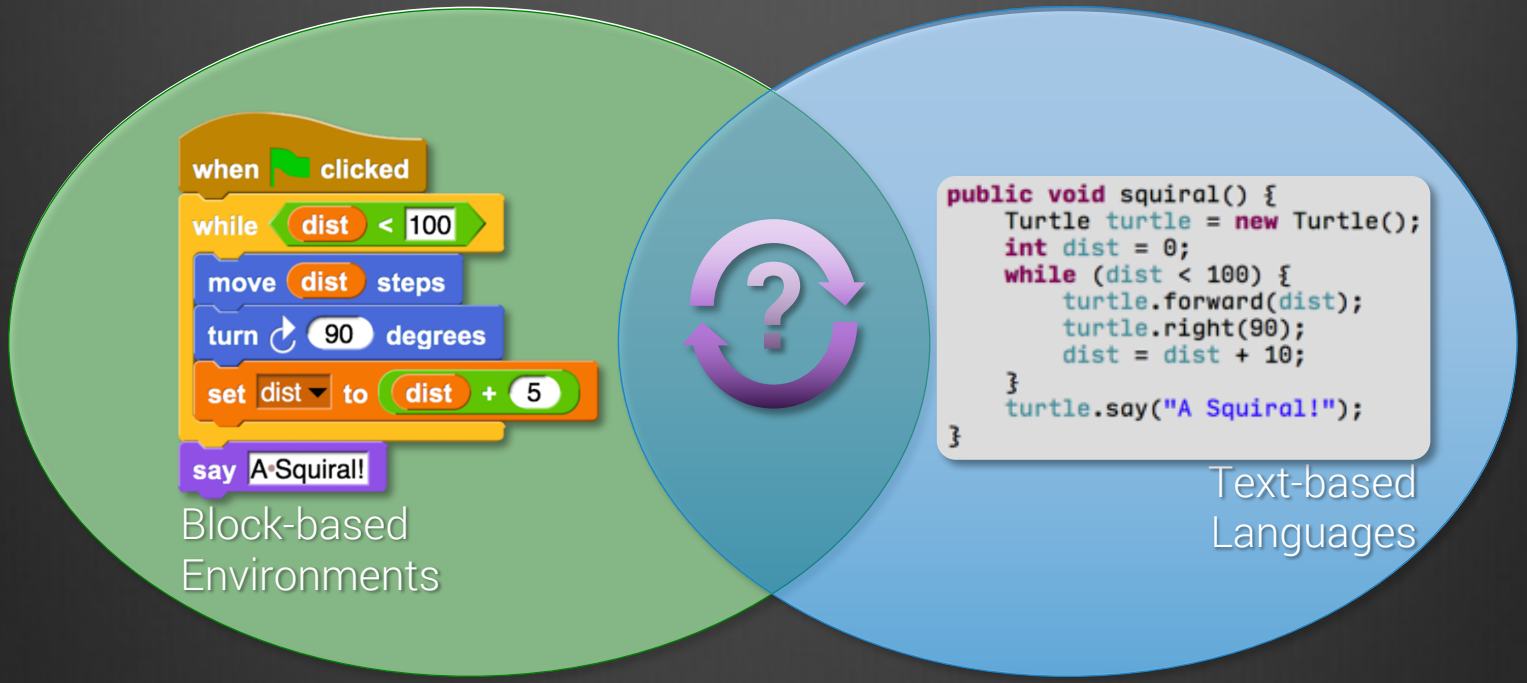


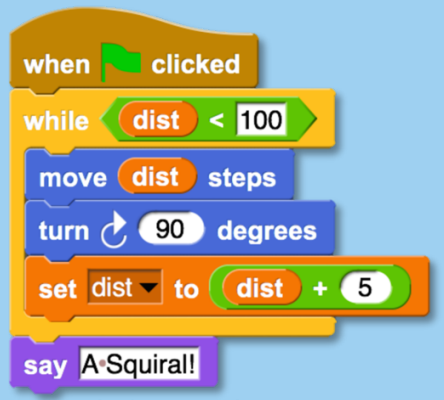

Block-based Environments

VS

```
public void squiral() {
    Turtle turtle = new Turtle();
    int dist = 0;
    while (dist < 100) {
        turtle.forward(dist);
        turtle.right(90);
        dist = dist + 10;
    }
    turtle.say("A Squiral!");
}
```

Text-based Languages


when clicked
while dist < 100
move dist steps
turn 90 degrees
set dist to dist + 5
say A Squiral!
Block-based Environments
public void squiral() {
Turtle turtle = new Turtle();
int dist = 0;
while (dist < 100) {
turtle.forward(dist);
turtle.right(90);
dist = dist + 10;
}
turtle.say("A Squiral!");
}
Text-based Languages



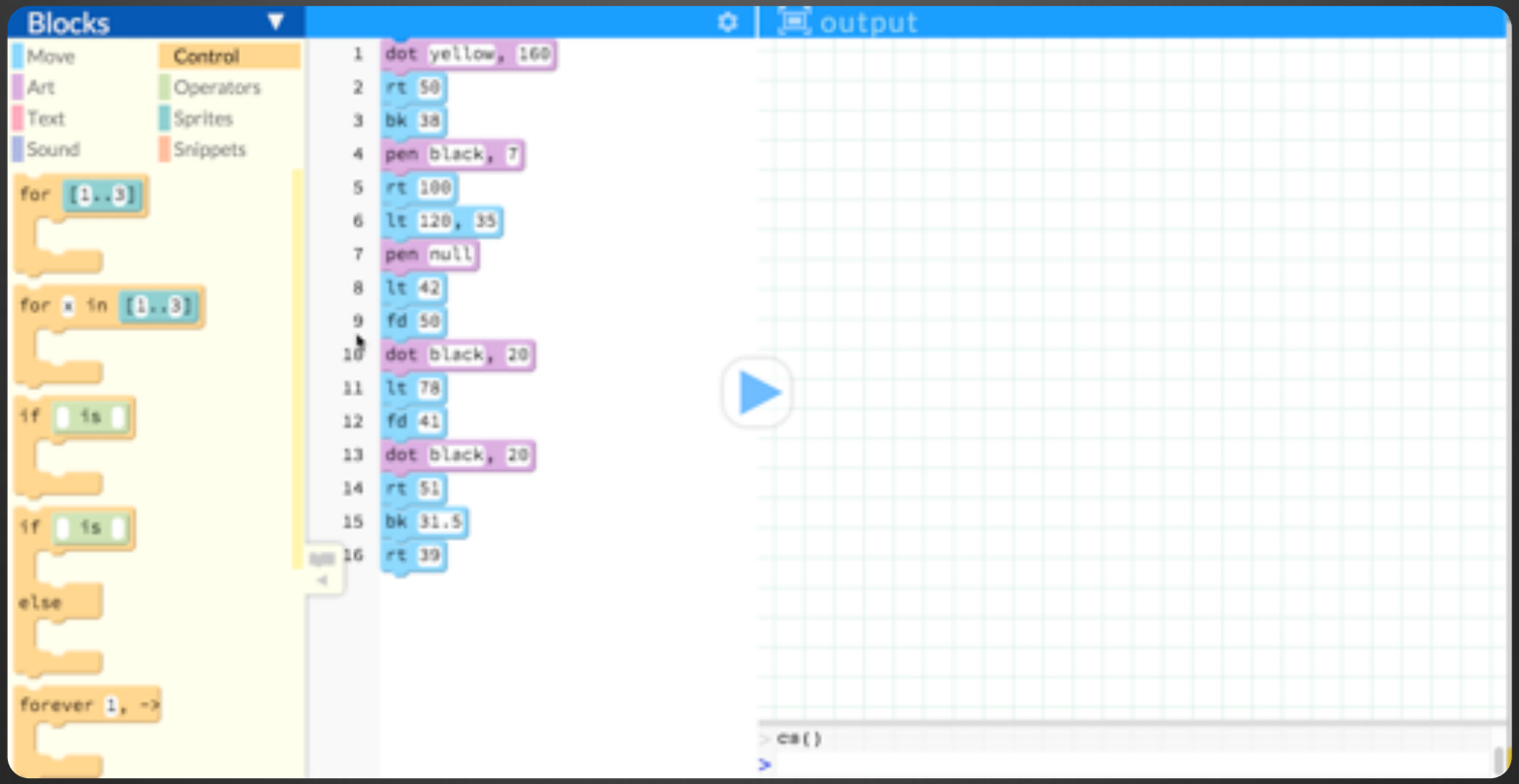

Block-based Environments

&

```
public void squiral() {
    Turtle turtle = new Turtle();
    int dist = 0;
    while (dist < 100) {
        turtle.forward(dist);
        turtle.right(90);
        dist = dist + 10;
    }
    turtle.say("A Squiral!");
}
```

Text-based Languages

# Dual-Modality Environments




[SIGCSE - Weintrop & Holbert, 2017]

# The Role of Block-based Programming In Computer Science Education

David Weintrop
weintrop@umd.edu
University of Maryland

Raspberry Pi CER Seminar
December 1 , 2020