

CSE 390B, Spring 2022

Building Academic Success Through Bottom-Up Computing

# Test Taking Strategies & Midterm Practice Exam

Test Taking Strategies, Midterm Practice Exam, Mock Exam  
Walkthrough, Revisiting the Metacognition Cycle

# Lecture Outline

- ❖ **Test Taking Strategies**
  - **Maximizing Success on Exam Day**
- ❖ **Midterm Practice Exam**
  - **Mock Exam, Debrief, and Reflection**
- ❖ **Midterm Practice Exam Walkthrough**
  - **Solutions and Exploring Sample Rubrics**
- ❖ **Revisiting the Metacognition Cycle**
  - **Applying Metacognition to Exams**

# Test Taking Strategies Discussion

- ❖ What are some test taking strategies you have previously utilized in taking your exams?
- ❖ Were those strategies you tried effective or not? Why?
- ❖ How might you try a new test taking strategy on the CSE 390B midterm or any other upcoming exam?

# Test Taking Best Practices

- ❖ Survey the entire exam before beginning
  - Helps plan how much time to allocate for each problem
- ❖ Read exam directions and question statements carefully
  - Use **highlights**, underlines, circles on important details
- ❖ Answer the questions you feel the most confident in first
- ❖ If stuck on a problem, make a mark on the problem and revisit the question later

# Test Taking Best Practices

- ❖ Prioritize how you will answer questions
  - Do this based on confidence level for each type of question or how long you think each will take
- ❖ Rely on a methodological approach for each question
  - Helps make taking the test feel more systematic
- ❖ If stuck on a question, demonstrate what you know
  - Many exams reward partial credit
- ❖ If time allows, double check your answers
  - Catches any small mistakes that may have been made earlier

# Lecture Outline

- ❖ Test Taking Strategies
  - Maximizing Success on Exam Day
- ❖ **Midterm Practice Exam**
  - **Mock Exam, Debrief, and Reflection**
- ❖ Midterm Practice Exam Walkthrough
  - Solutions and Exploring Sample Rubrics
- ❖ Revisiting the Metacognition Cycle
  - Applying Metacognition to Exams

# Midterm Practice Exam

- ❖ The exam is closed-note, closed-book
  - You may only use the midterm reference sheet
- ❖ Questions are not necessarily in order of difficulty
- ❖ You will have 25 minutes to complete the mock exam
  - We will give you a 5-minute warning
- ❖ Remember to relax and take deep breaths

# Mock Exam Debrief & Reflection

- ❖ What did you learn about yourself through this process?  
About your test taking practices?
- ❖ What are two test taking strategies that you would like to engage with in your next exam? Why?
- ❖ What is one thing that can help you relax and calm down before or during your exam?

# Five-minute Break!

- ❖ Feel free to stand up, stretch, use the restroom, drink some water, review your notes, or ask questions
- ❖ We'll be back at:
- ❖ Research shows mid-lecture breaks reduce the decline of attention in the middle of lecture (Olmsted, 1999)

# Lecture Outline

- ❖ Test Taking Strategies
  - Maximizing Success on Exam Day
- ❖ Midterm Practice Exam
  - Mock Exam, Debrief, and Reflection
- ❖ **Midterm Practice Exam Walkthrough**
  - **Solutions and Exploring Sample Rubrics**
- ❖ Revisiting the Metacognition Cycle
  - Applying Metacognition to Exams

# Question 1: Circuit Design

## Part a: Truth Table

$A_t$	$B_t$	$A_{t+1}$	$B_{t+1}$
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

11  $\rightarrow$  10  $\rightarrow$  01  $\rightarrow$  00  $\rightarrow$  11

# Question 1: Circuit Design

## Part a: Truth Table

$A_t$	$B_t$	$A_{t+1}$	$B_{t+1}$
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

11 → 10 → 01 → 00 → 11

# Question 1: Circuit Design

## Part a: Truth Table

$A_t$	$B_t$	$A_{t+1}$	$B_{t+1}$
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

$$A_{t+1} = (A_t \& B_t)$$

$$A_{t+1} = (\sim A_t \& \sim B_t)$$

## Part b: Boolean Expressions

# Question 1: Circuit Design

## Part a: Truth Table

$A_t$	$B_t$	$A_{t+1}$	$B_{t+1}$
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

$$A_{t+1} = (A_t \& B_t)$$

$$A_{t+1} = (\sim A_t \& \sim B_t)$$

## Part b: Boolean Expressions

$$A_{t+1} = (A_t \& B_t) \mid (\sim A_t \& \sim B_t)$$

# Question 1: Circuit Design

## Part a: Truth Table

$A_t$	$B_t$	$A_{t+1}$	$B_{t+1}$
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

$$A_{t+1} = (A_t \& B_t)$$

$$B_{t+1} = (A_t \& \sim B_t)$$

$$A_{t+1} = (\sim A_t \& \sim B_t)$$

$$B_{t+1} = (\sim A_t \& \sim B_t)$$

## Part b: Boolean Expressions

$$A_{t+1} = (A_t \& B_t) \mid (\sim A_t \& \sim B_t)$$

$$B_{t+1} = (A_t \& \sim B_t) \mid (\sim A_t \& \sim B_t)$$

$$= \sim B_t \& (A_t \mid \sim A_t)$$

[Factor out  $\sim B_t$ ]

$$= \sim B_t$$

[ $A_t \mid \sim A_t = 1$ ]

# Question 1: Circuit Design

## Part a: Truth Table

$A_t$	$B_t$	$A_{t+1}$	$B_{t+1}$
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

## Part c: Drawing the Circuit

## Part b: Boolean Expressions

$$A_{t+1} = (A_t \& B_t) \mid (\sim A_t \& \sim B_t)$$

$$B_{t+1} = (A_t \& \sim B_t) \mid (\sim A_t \& \sim B_t)$$

$$= \sim B_t \& (A_t \mid \sim A_t)$$

$$= \sim B_t$$

[Factor out  $\sim B_t$ ]

[ $A_t \mid \sim A_t = 1$ ]

# Question 1: Circuit Design Sample Rubric

Category	Points	Criteria
Truth Table	4 points	1 point for each row in the truth table that is correct
Boolean Expressions	6 points	<ul style="list-style-type: none"><li>❖ 4 points for correct expression for <math>A_{t+1}</math><ul style="list-style-type: none"><li>▪ 2 points if truth table is wrong but expression matches truth table</li></ul></li><li>❖ 2 points for correct expression for <math>B_{t+1}</math><ul style="list-style-type: none"><li>▪ 1 point if truth table is wrong but expression matches truth table</li></ul></li></ul>
Circuit Diagram	5 points	<ul style="list-style-type: none"><li>❖ 3 points for having circuits that match the Boolean expressions in part b</li><li>❖ 2 points for fully correct diagram</li></ul>
<b>Total</b>	<b>15 points</b>	

## Question 2: Math Puzzle

Dana needs 300 pickets for her colorful picket fence. She wants equal amounts of each of her 4 selected colors. She already has 32 red, 26 green, 9 yellow, and no blue. If the pickets cost 25 cents and you get 20% off if you purchase 50 or more of the same color, and 30% off if you purchase 60 or more of one color, how much does Dana need to spend? List your answer to two decimal places. You may use a calculator application on your computer to solve this problem.

## Question 2: Math Puzzle

Dana needs 300 pickets for her colorful picket fence. She wants equal amounts of each of her 4 selected colors. She already has 32 red, 26 green, 9 yellow, and no blue. If the pickets cost 25 cents and you get 20% off if you purchase 50 or more of the same color, and 30% off if you purchase 60 or more of one color, how much does Dana need to spend? List your answer to two decimal places. You may use a calculator application on your computer to solve this problem.

### Solution

$$75 - 32 = 43 \text{ red}$$

$$75 - 26 = 49 \text{ green}$$

$$75 - 9 = 66 \text{ yellow}$$

$$75 - 0 = 75 \text{ blue}$$

$$43 \times 0.25 + 49 \times 0.25 + 0.7 \times 66 \times 0.25 + 0.7 \times 75 \times 0.25$$

$$= \$47.675$$

$$= \$47.68 \quad (\text{Rounding down is also acceptable})$$

# Question 3: Hack Assembly Programming

Write a Hack assembly program that stores -1, 0, or 1 in R1 based on the sign of R0. To be more specific, your program should store a -1 in R1 if R0 is negative, a 0 in R1 if R0 is 0, and a 1 in R1 if R0 is positive.

j1 ( <i>out</i> < 0)	j2 ( <i>out</i> = 0)	j3 ( <i>out</i> > 0)	Mnemonic	Effect
0	0	0	null	No jump
0	0	1	JGT	If <i>out</i> > 0 jump
0	1	0	JEQ	If <i>out</i> = 0 jump
0	1	1	JGE	If <i>out</i> ≥ 0 jump
1	0	0	JLT	If <i>out</i> < 0 jump
1	0	1	JNE	If <i>out</i> ≠ 0 jump
1	1	0	JLE	If <i>out</i> ≤ 0 jump
1	1	1	JMP	Jump

# Question 3: Hack Assembly Programming

Write a Hack assembly program that stores -1, 0, or 1 in R1 based on the sign of R0. To be more specific, your program should store a -1 in R1 if R0 is negative, a 0 in R1 if R0 is 0, and a 1 in R1 if R0 is positive.

## Equivalent pseudocode:

```
if (R0 < 0) {
    R1 = -1;
} else if (R0 > 0) {
    R1 = 1;
} else { //R0 == 0
    R1 = 0;
}
```

j1 ( <i>out</i> < 0)	j2 ( <i>out</i> = 0)	j3 ( <i>out</i> > 0)	Mnemonic	Effect
0	0	0	null	No jump
0	0	1	JGT	If <i>out</i> > 0 jump
0	1	0	JEQ	If <i>out</i> = 0 jump
0	1	1	JGE	If <i>out</i> ≥ 0 jump
1	0	0	JLT	If <i>out</i> < 0 jump
1	0	1	JNE	If <i>out</i> ≠ 0 jump
1	1	0	JLE	If <i>out</i> ≤ 0 jump
1	1	1	JMP	Jump

# Question 3: Hack Assembly Programming

Write a Hack assembly program that stores -1, 0, or 1 in R1 based on the sign of R0. To be more specific, your program should store a -1 in R1 if R0 is negative, a 0 in R1 if R0 is 0, and a 1 in R1 if R0 is positive.

## Equivalent pseudocode:

```
if (R0 < 0) {
    R1 = -1;
} else if (R0 > 0) {
    R1 = 1;
} else { //R0 == 0
    R1 = 0;
}
```

j1 (out < 0)	j2 (out = 0)	j3 (out > 0)	Mnemonic	Effect
0	0	0	null	No jump
0	0	1	JGT	If out > 0 jump
0	1	0	JEQ	If out = 0 jump
0	1	1	JGE	If out ≥ 0 jump
1	0	0	JLT	If out < 0 jump
1	0	1	JNE	If out ≠ 0 jump
1	1	0	JLE	If out ≤ 0 jump
1	1	1	JMP	Jump

One solution:

```
@R0
D = M
@NEGATIVE
D; JLT
@POSITIVE
D; JGT
// R0 == 0 case
@R1
M = 0
@END
0; JMP
(NEGATIVE)
// R0 < 0 case
@R1
M = -1
@END
0; JMP
(POSITIVE)
// R0 > 0 case
@R1
M = 1
(END)
@END
0; JMP
```

# Question 3: Hack Assembly Sample Rubric

Category	Points	Criteria
Has Infinite End Loop	1 point	❖ 1 point if program has an Infinite End Loop
Conditional Checks	4 points	<ul style="list-style-type: none"><li>❖ 2 points for having at least two checks for cases. Almost all solutions will need a check for 2 of the three cases (negative, zero, positive).</li><li>❖ 2 points for correctly matching jump condition to cases (e.g. jump to negative case when negative, etc.)</li></ul>
Assigns Correct R1 Value	3 points	<ul style="list-style-type: none"><li>❖ One point for each case:<ul style="list-style-type: none"><li>▪ negative: <math>R1 = -1</math></li><li>▪ zero: <math>R1 = 0</math></li><li>▪ positive: <math>R = 1</math></li></ul></li></ul>
Fully Correct Implementation	2 points	❖ Covers any little mistakes that may result in an incorrect implementation (e.g., forgetting to jump to the end when a case is done)
<b>Total</b>	<b>10 points</b>	

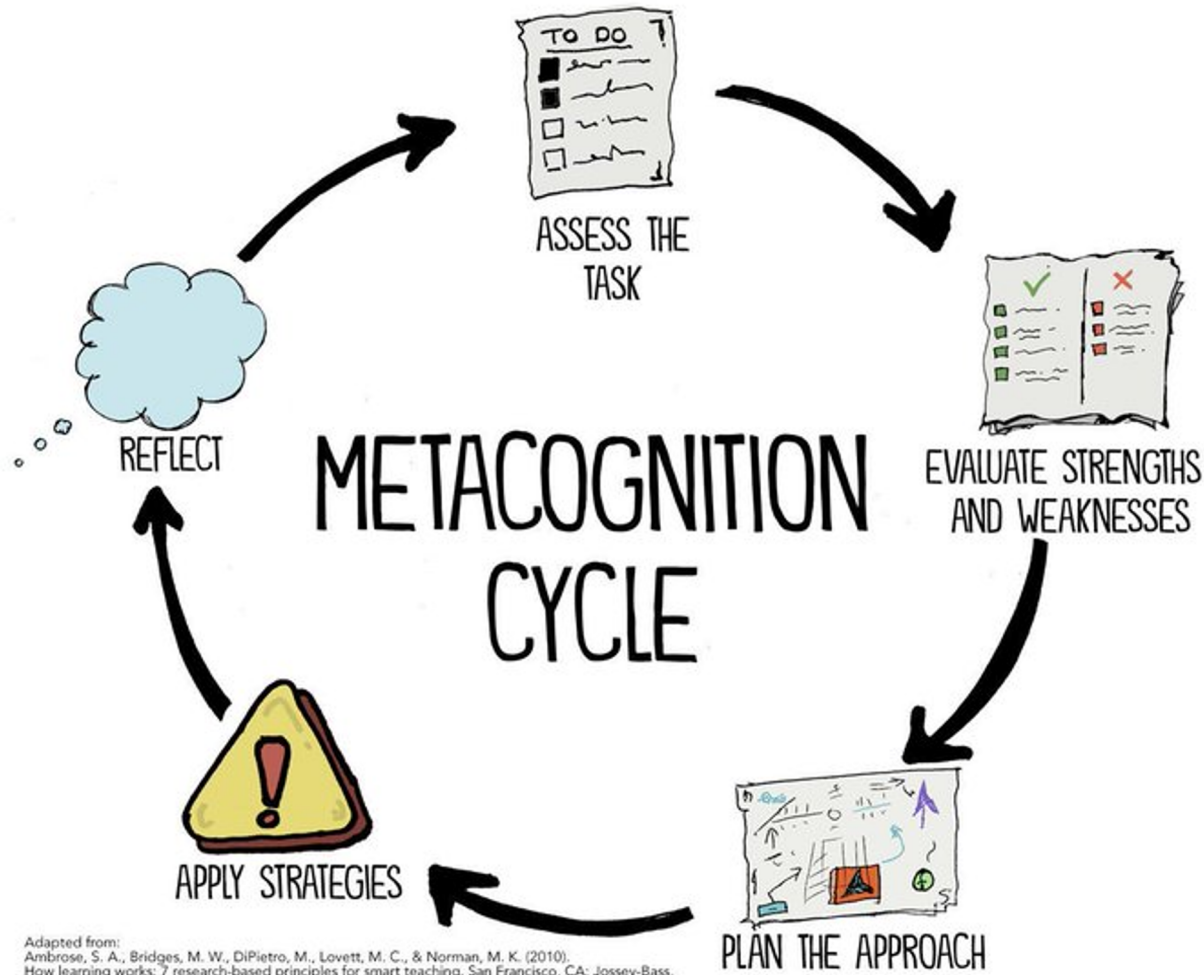
# Question 4: Metacognitive Skills

- ❖ Name two metacognitive skills that we have covered in CSE 390B thus far.

# Lecture Outline

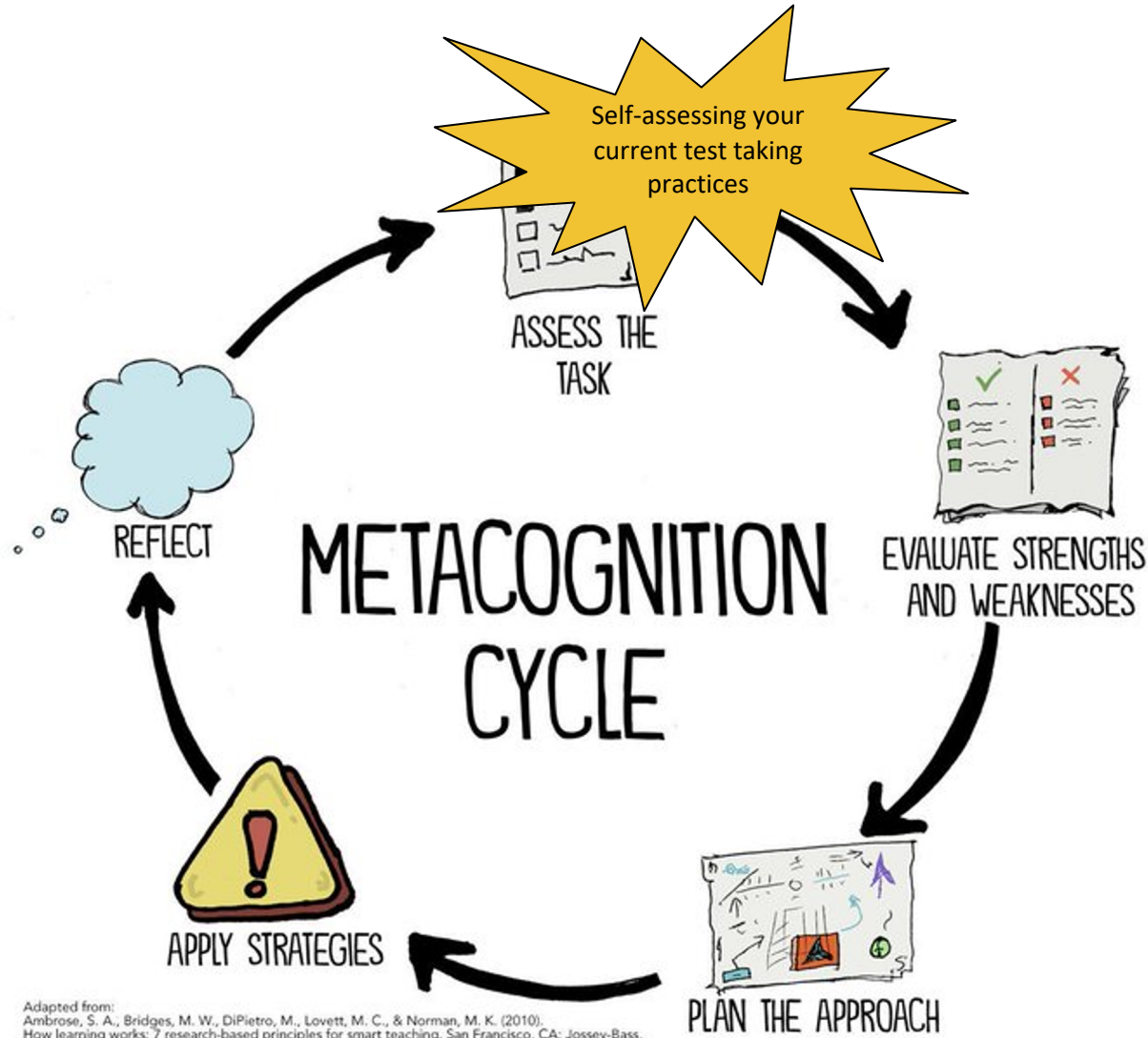
- ❖ Test Taking Strategies
  - Maximizing Success on Exam Day
- ❖ Midterm Practice Exam
  - Mock Exam, Debrief, and Reflection
- ❖ Midterm Practice Exam Walkthrough
  - Solutions and Exploring Sample Rubrics
- ❖ **Revisiting the Metacognition Cycle**
  - **Applying Metacognition to Exams**

# Revisiting the Metacognition Cycle



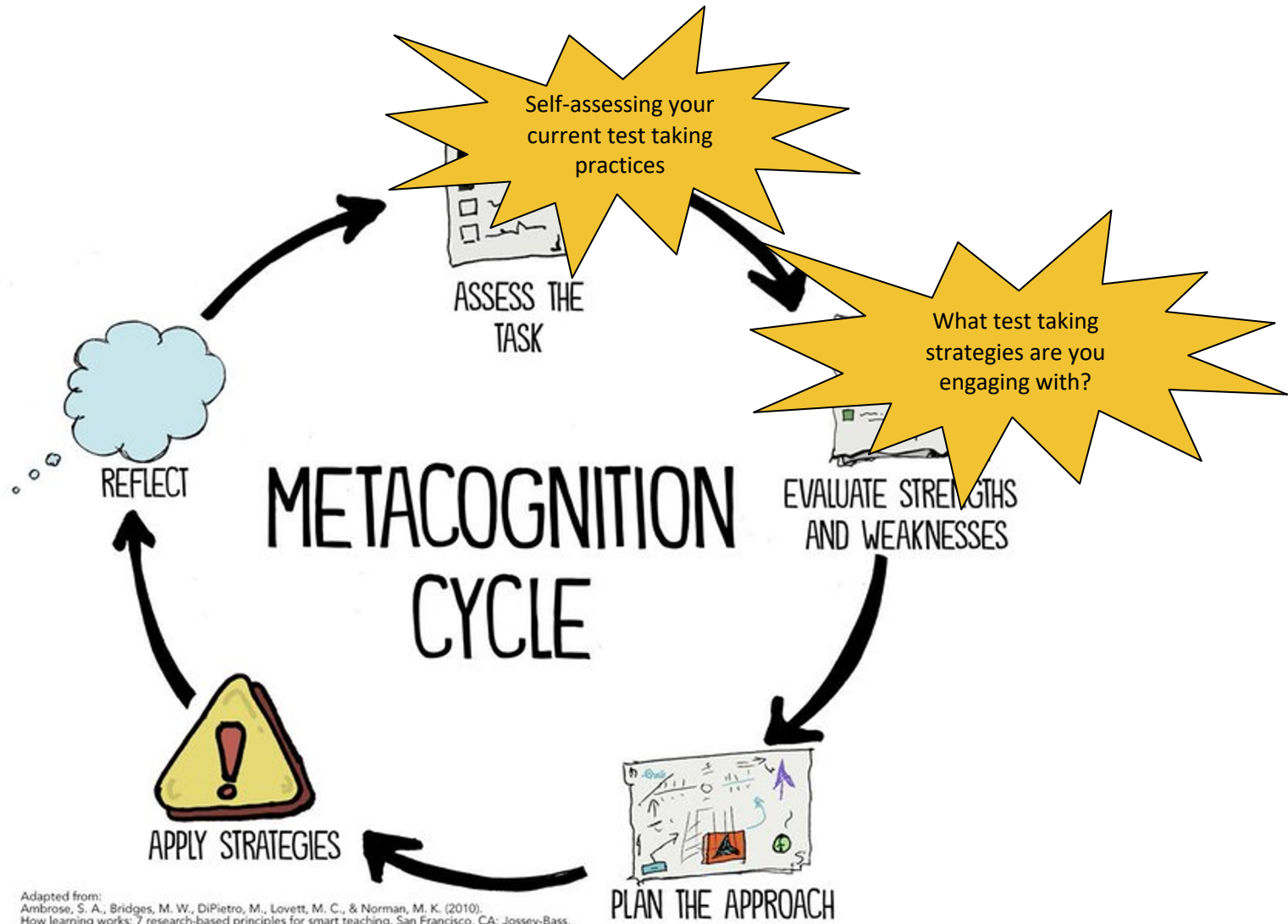
Adapted from:  
 Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010).  
 How learning works: 7 research-based principles for smart teaching. San Francisco, CA: Jossey-Bass.

# Revisiting the Metacognition Cycle



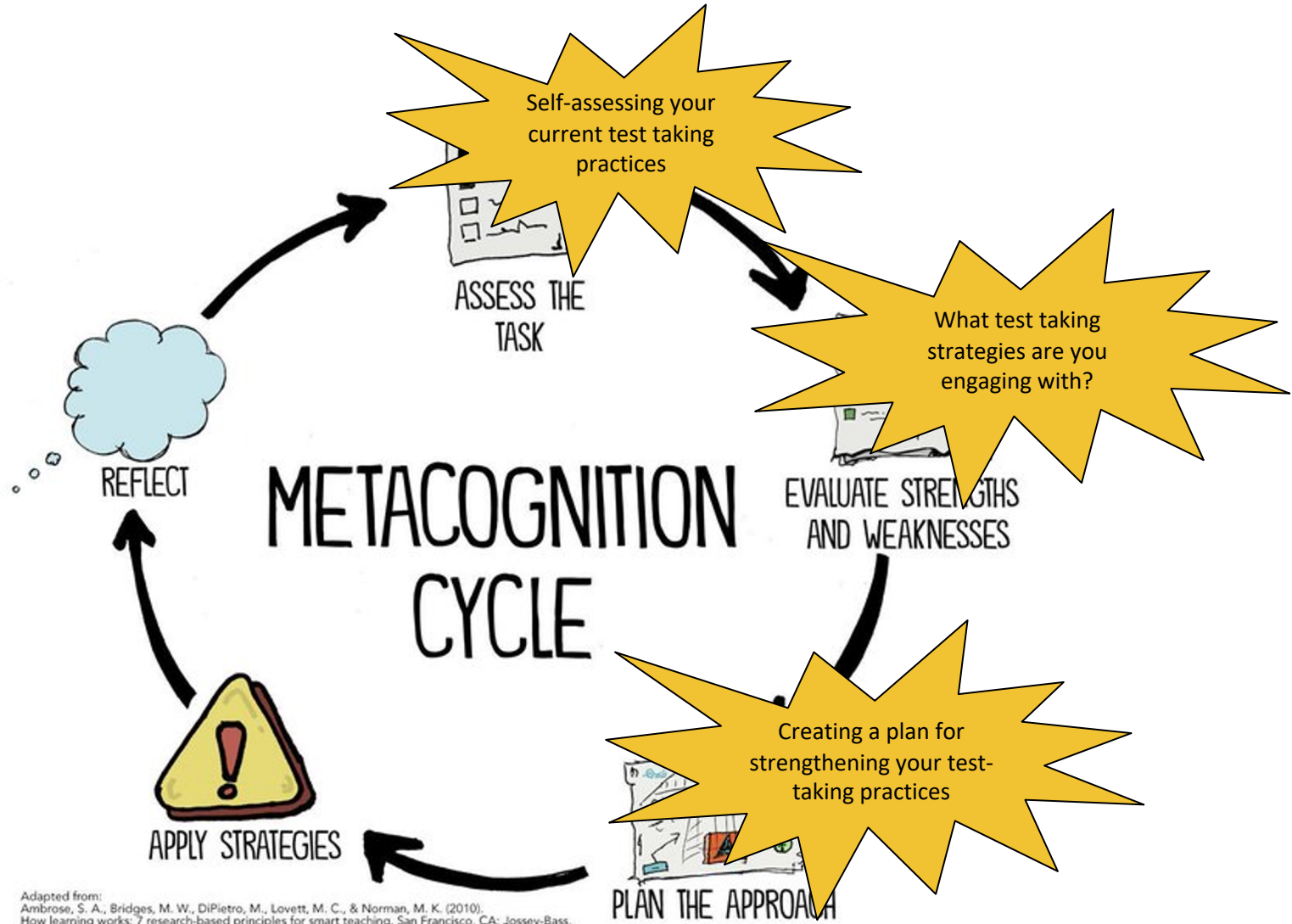
Adapted from:  
 Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010).  
 How learning works: 7 research-based principles for smart teaching. San Francisco, CA: Jossey-Bass.

# Revisiting the Metacognition Cycle



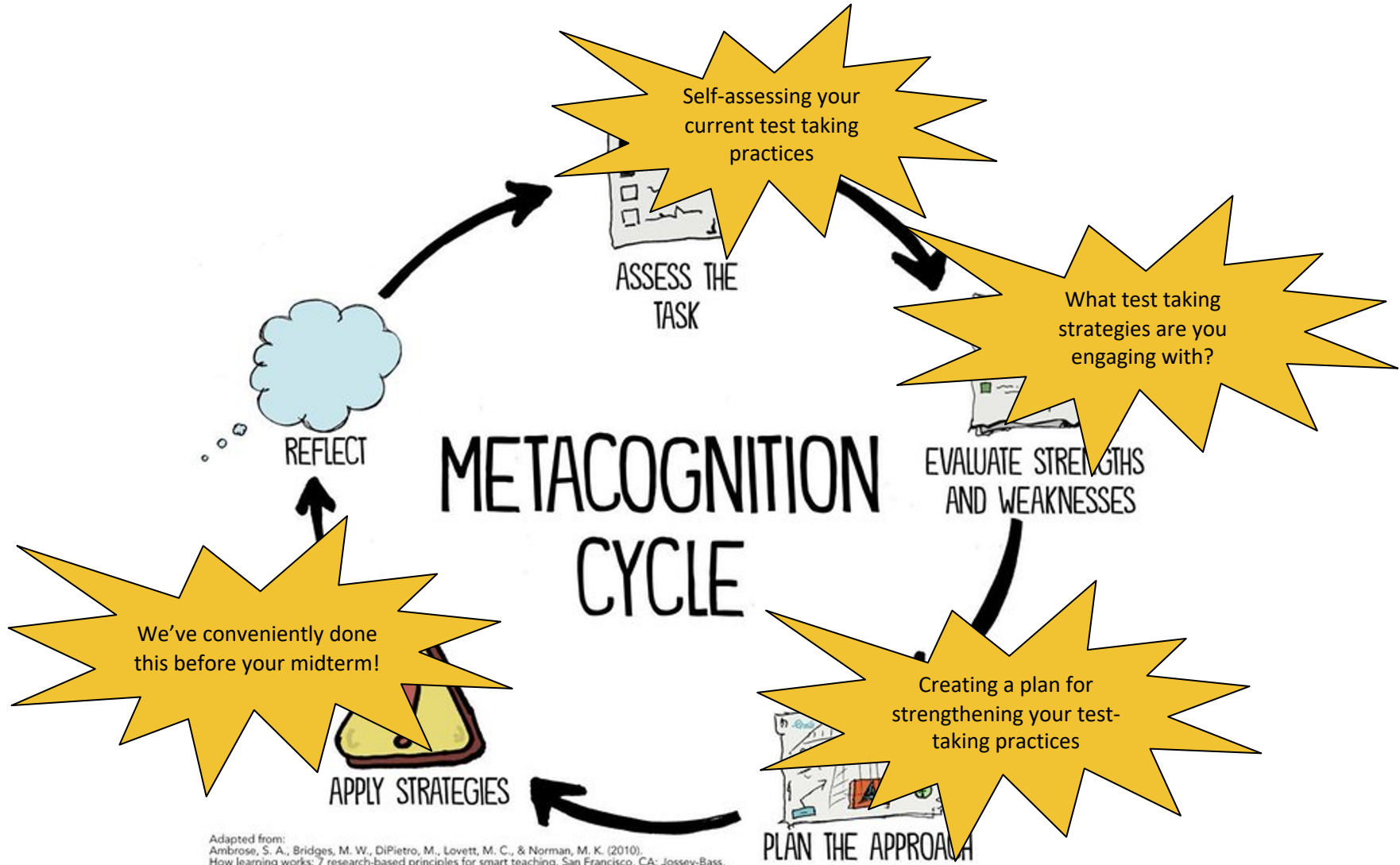
Adapted from:  
 Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010).  
 How learning works: 7 research-based principles for smart teaching. San Francisco, CA: Jossey-Bass.

# Revisiting the Metacognition Cycle



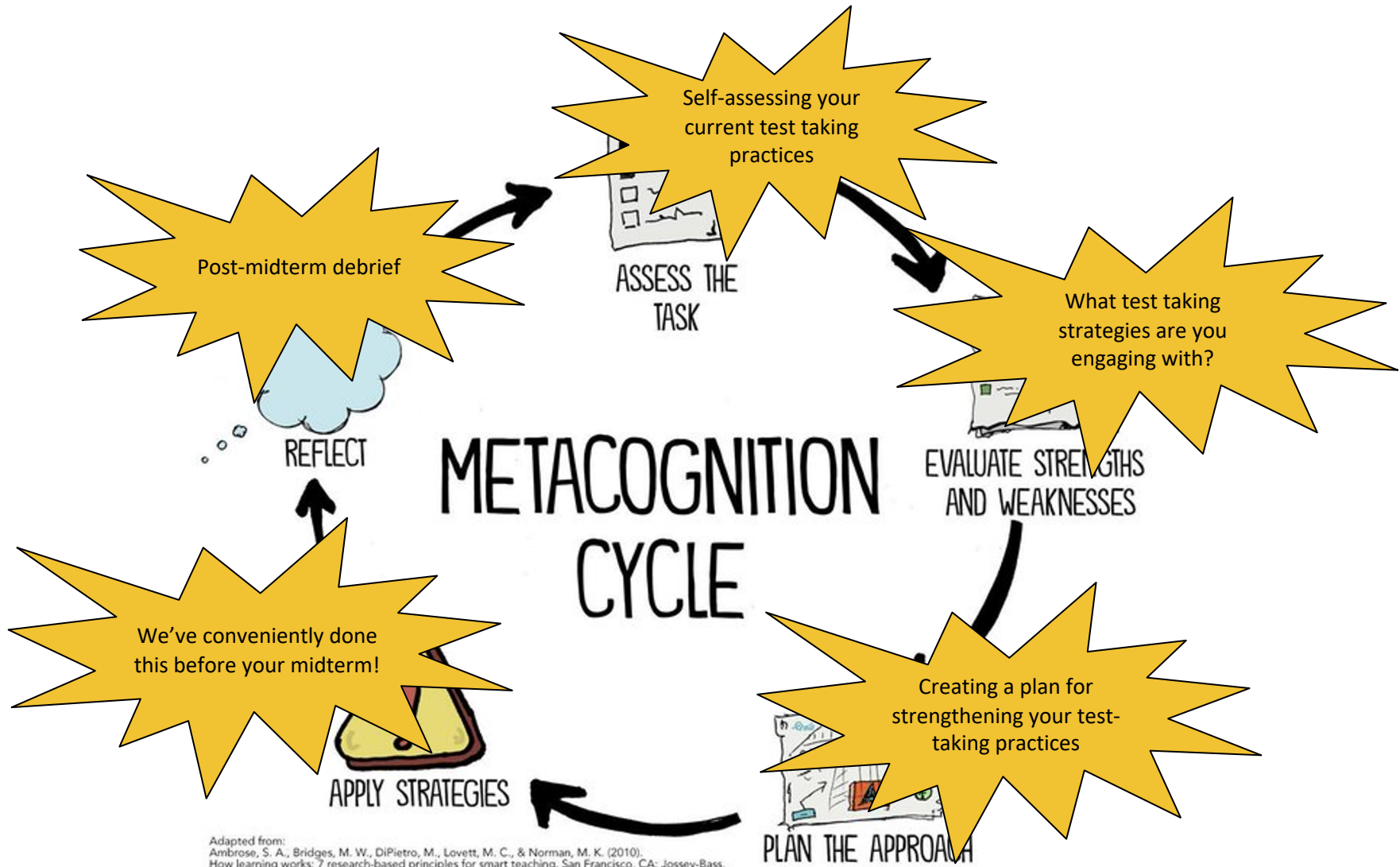
Adapted from:  
 Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010).  
 How learning works: 7 research-based principles for smart teaching. San Francisco, CA: Jossey-Bass.

# Revisiting the Metacognition Cycle



Adapted from:  
Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010).  
How learning works: 7 research-based principles for smart teaching. San Francisco, CA: Jossey-Bass.

# Revisiting the Metacognition Cycle



Adapted from:  
 Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010).  
 How learning works: 7 research-based principles for smart teaching. San Francisco, CA: Jossey-Bass.

# Lecture 11 Wrap-up

- ❖ CSE 390B Midterm Exam this Thursday (5/5) during lecture at 2:30pm
- ❖ Project 6: Mock Exam Problem & Building a Computer due next Thursday (5/12) at 11:59pm PDT