

**Q: What is the value of “age” after all lines of this program are executed?**



- A. 1
- B. 40
- C. 41
- D. 401

```

when clicked
  set age to 40
  say join Your age is now age for 2 secs
  set age to age + 1
  say join Happy Birthday! Your age is now age for 2 secs
  
```



# CPSC 100

# Computational Thinking

## Intro to Programming

**Instructor: Parsa Rajabi**  
**Department of Computer Science**  
**University of British Columbia**



# Agenda

- Learning Goals
- Course Admin
- Intro to Programming [Continued]



# Learning Goals

After this week's lecture, you should be able to:

- Identify the differences between sequential and "breaking bad" algorithms
- Discuss the **difference** between high level, assembly & machine code.
- **Identify and describe** the components of an algorithm
  - (i.e., sequencing, selection, and iteration)
- **Use snap blocks** to represent algorithms
- Be able to **trace** through code using sequences of instructions, variables, loops, and conditional statements in short programs
  - Read carefully: it says be able to **trace** code, **not write code**. In order to help you do this, you will write a small amount of code in lab. You will not, however, be asked to write code on exam.
- Describe in English what a block of *Snap!* code does.

# Course Admin



# Course Admin

- **Lab #3**
  - Late submission available until Friday, Jan 31 at 11:59pm
- **Post-Class (PC) Quiz #2**
  - Only 1 attempt, 60 minutes
  - Due on Sunday, Feb 2 at 11:59pm
- **Post-Class (PC) Quiz #3**
  - To be available starting Monday
- **Project**
  - Milestone 1 - Proposal (5%) **Feb. 12**
  - You should have started by now! If not, start **today**.

# TA

# Office Hours



2024W2\_V

[Home](#)

[Announcements](#)

[Syllabus](#)

**[Contact Parsa or TAs](#)**

[Discussion Board](#)  
[\[Ed\]](#)

[Project Details](#)

[Assignments](#)

[People](#)

[iClicker Cloud](#)


[Grades](#)

## Teaching Team

### Instructor

Name	Email 	Office Hours 	Location
Parsa Rajabi	prajabi [at] <del>DELETethisTEXT</del> cs.ubc.ca	Wednesdays 1:30-2:30	ICCS 249

### Teaching Assistants

Lab Section	Name	Email 	Book Office Hours
L2A (9-10a)	<b>Olamide</b> Olabiyi	oolabiyi [at] <del>DELETethisTEXT</del> student.ubc.ca	<a href="#">Booking Link</a>
L2B (10-11a)	<b>Parsa</b> Seyfourian	parsasey [at] <del>DELETethisTEXT</del> student.ubc.ca	<a href="#">Booking Link</a>
L2C (11-12p)	<b>Kelly</b> Xi	kellyxi [at] <del>DELETethisTEXT</del> student.ubc.ca	<a href="#">Booking Link</a>
L2D (12-1p)	<b>Abigail</b> Demian	ademio01 [at] <del>DELETethisTEXT</del> student.ubc.ca	-
L2E (5-6p)	<b>Sam</b> Bakteria	bakteria [at] <del>DELETethisTEXT</del> student.ubc.ca	<a href="#">Booking Link</a>





# Snow day?



- We will keep a very close eye on the weather
- UBC/CS has a process in place and we shall follow their guidance
- You should keep a close eye on **canvas announcements**
- In case it snows AND UBC is still open...
  - We will use our best judgement
  - I will inform you of **any changes by Monday, at 9:30am**
  - By default, we **WILL** have class, unless stated otherwise
- Top priority: **safety**! Take care of yourself first, then help others.

# Mutation

Process of **changing the state** or data of an **object** after it has been created.

**Repeat 10 times:**

1. Preheat oven (400° C)
2. **Combine ingredients in bowl to form dough**
3. Put dough into bread pan
4. If ingredients contain yeast, allow to sit at room temperature for 1 hour
5. Put bread pans into preheated oven and bake for 30 minutes



# Components of an Algorithm

1. Sequencing
2. Selection
3. Iteration

# Components of an Algorithm

1. **Sequencing**
2. **Selection**
3. **Iteration**



# Sequencing

Instructions are executed in the specified order



**Repeat 10 times:**

1. Preheat oven (400° C)
2. Combine ingredients in bowl to form dough
3. Put dough into bread pan
4. If ingredients contain yeast, allow to sit at room temperature for 1 hour
5. Put bread pans into preheated oven and bake for 30 minutes





# Sequencing

## Order matters



Programs will execute exactly in the order that's given:

1. A
2. B
3. C

If we assign values to variables, they'll set one value after another after another.

# Components of an Algorithm

- ~~1. Sequencing~~
- 2. Selection**
3. Iteration



# Selection

Allows the algorithm to select which instructions to execute  
(depending on conditions)



**Repeat 10 times:**

1. Preheat oven (400° C)
2. Combine ingredients in bowl to form dough
3. Put dough into bread pan
4. **If ingredients contain yeast, allow to sit at room temperature for 1 hour**
5. Put bread pans into preheated oven and bake for 30 minutes



# Selection → Conditionals

Conditionals allow for different results, depending on input. Generally this looks like “**if**” with a possible “**else**”:

Real World:

**If** you eat your dinner

***Then*** you may have some ice cream

**Else** (you may have referred to this step as "otherwise" in Lab 2)

You may only have fruit



# Selection → Conditionals

**If** ingredients contains yeast  
allow to sit at room temp. for 1hr



**If** it's snowing  
Wake-up 6:30am  
**Else**  
Wake-up at 7am



# Q: What's the output after we press run?



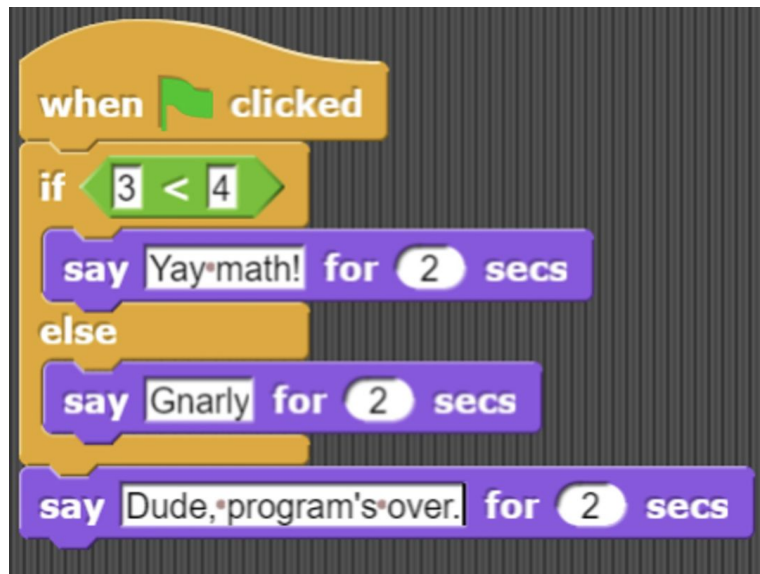
iClicker

A. It only says “Yay math!”

B. It says “Yay math!” Then...  
it says “Gnarly” Then...  
it says “Dude, program’s over.”

C. It says “Gnarly” Then...  
it says “Dude, program’s over.”

D. It says something else

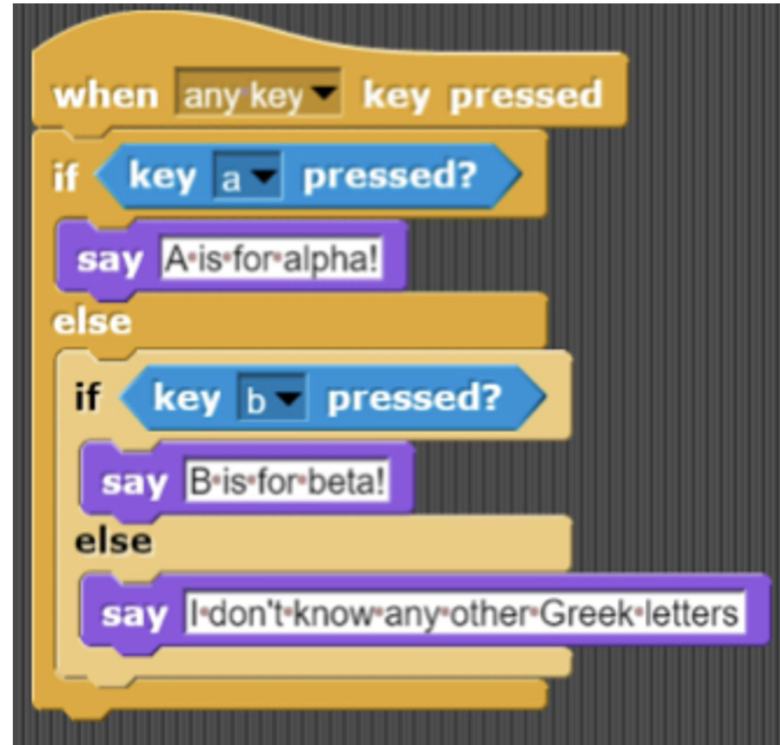


**Q: What is the output if you press the “P” key?**

- A. “A is for alpha”
- B. “B is for beta”
- C. “I don’t know any other Greek letters”
- D. The program crashes
- E. No output



iClicker





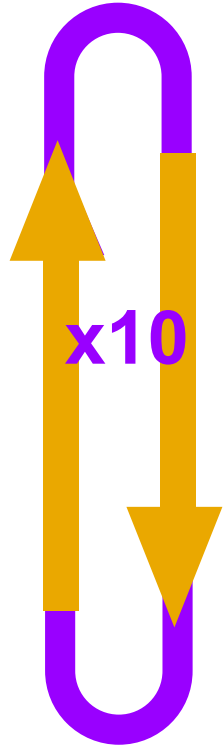


# Components of an Algorithm

- ~~1. Sequencing~~
- ~~2. Selection~~
- 3. Iteration**

# Iteration

Allows the algorithm to repeat instructions.



**Repeat 10 times:**

1. Preheat oven (400° C)
2. Combine ingredients in bowl to form dough
3. Put dough into bread pan
4. If ingredients contain yeast, allow to sit at room temperature for 1 hour
5. Put bread pans into preheated oven and bake for 30 minutes



7:35 📶 69%

**Otis Redding**

Music Merch

**you liked**  
1 song • Otis Redding

**Popular**

- 1 (Sittin' On) the Dock of the Bay 762,315,950
- 2 Hard to Handle 127,583,692
- 3 These Arms of Mine 124,408,002
- 4 Stand by Me 251,400,923
- 5 ... Try a Little Tenderness 148,756,793

**Artist Pick**

Posted by Otis Redding

**Try a Little Tenderness**  
Otis Redding

Home Search Your Library

7:11 📶 72%

**60s Mix**

**COMPLETE & UNBELIEVABLE THE OTIS REDDING DICTIONARY OF SOUL**

**MY-MY-MY**

**Try a Little Tenderness**  
Otis Redding

1:59 -1:22

🔍 ⏮ ⏪ ⏩ ⏭

🔍 ⏮ ⏪ ⏩ ⏭

Lyrics

7:11 📶 72%

**60s Mix**

**COMPLETE & UNBELIEVABLE THE OTIS REDDING DICTIONARY OF SOUL**

**MY-MY-MY**

**y a Little Tenderness**  
Otis Redding

1:59 -1:22

🔍 ⏮ ⏪ ⏩ ⏭

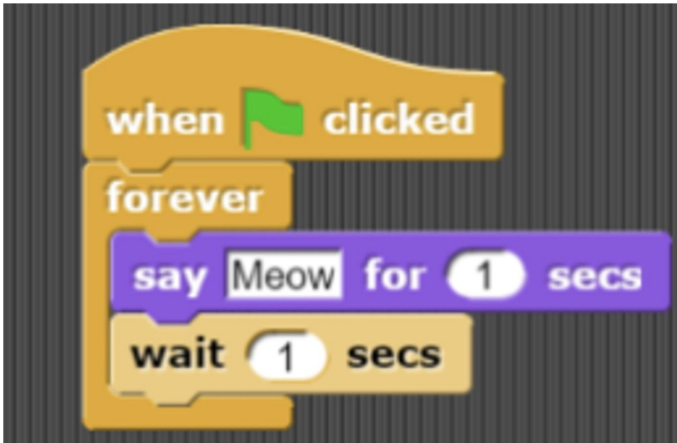
🔍 ⏮ ⏪ ⏩ ⏭

Lyrics

# Iteration

What if you want to do a task over and over again?

A **loop** allows you to do the same task over & over again, sometimes with a **stopping** condition, sometimes **forever**!

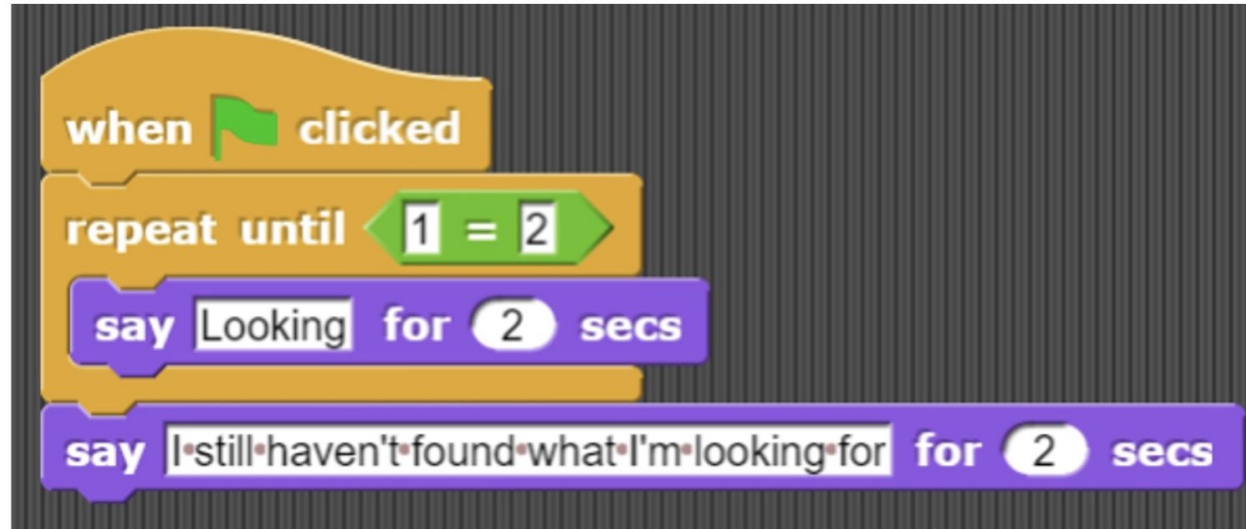




**Q: Will this program ever say “I still haven’t found what I’m looking for”?**



- A. Yes
- B. No
- C. Sometimes







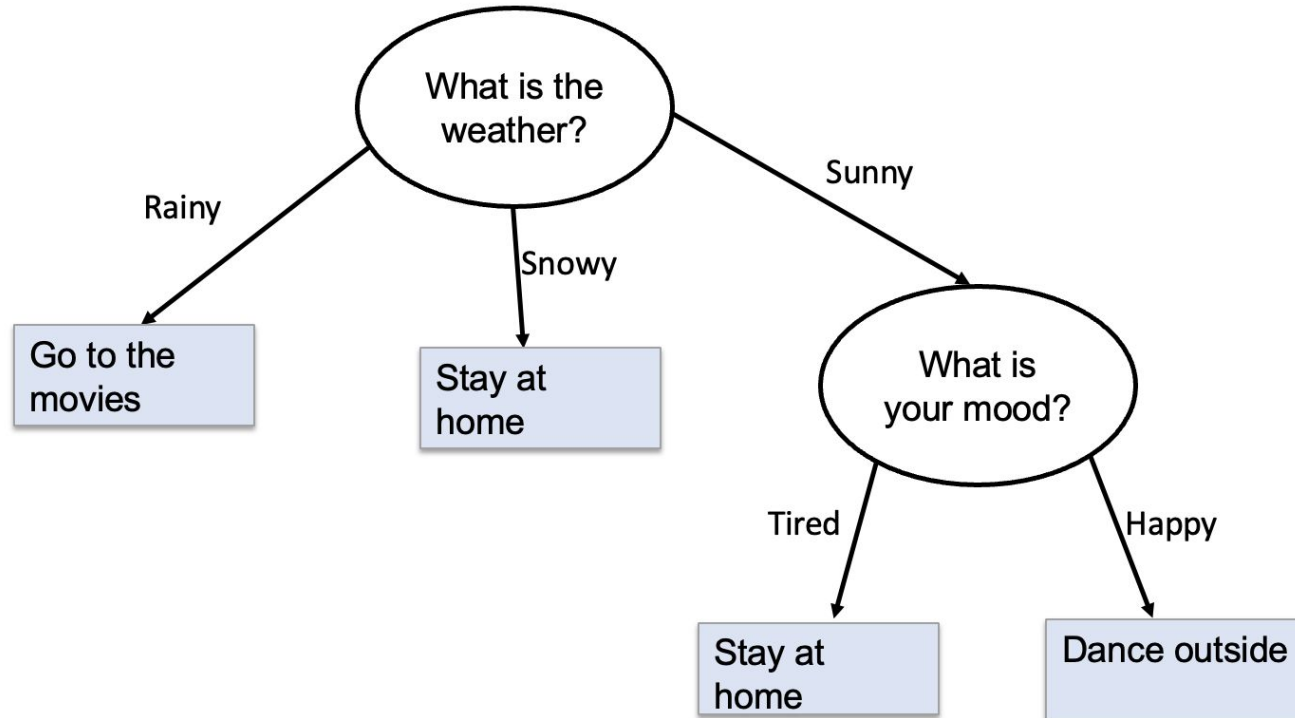
# Activity





# Activity: Conditionals in Snap!

Convert the following decision tree to a Snap Block program



# Wrap up



# Wrap Up

- **Lab #3**
  - Late submission available until Friday, Jan 31 at 11:59pm
- **Post-Class (PC) Quiz #2**
  - Only 1 attempt, 60 minutes
  - Due on Sunday, Feb 2 at 11:59pm
- **Post-Class (PC) Quiz #3**
  - To be available starting Monday
- **Project**
  - Milestone 1 - Proposal (5%) **Feb. 12**
  - You should have started by now! If not, start **today**.