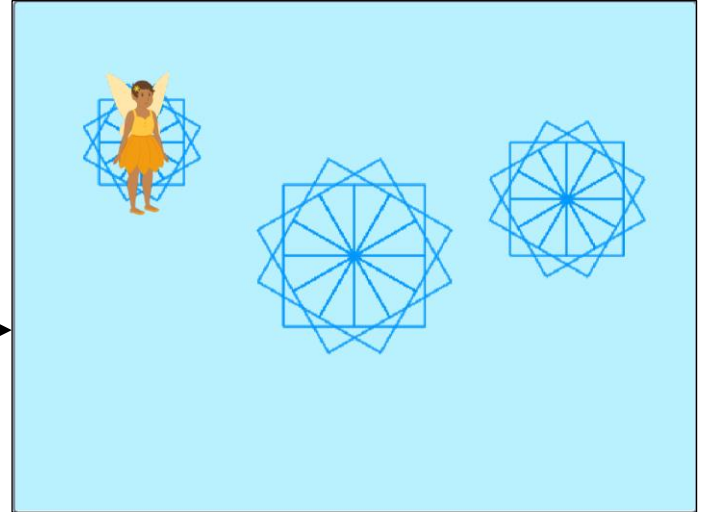


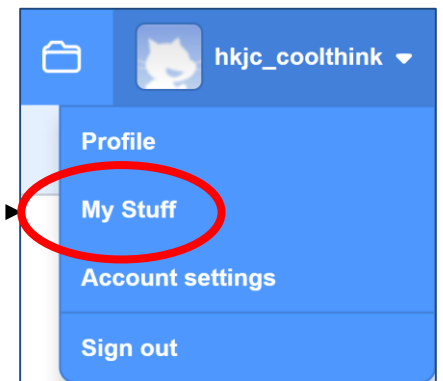
# COMPUTATIONAL ARTS WITH SCRATCH

In this lesson, you will change the size of your snowflakes in your art project.

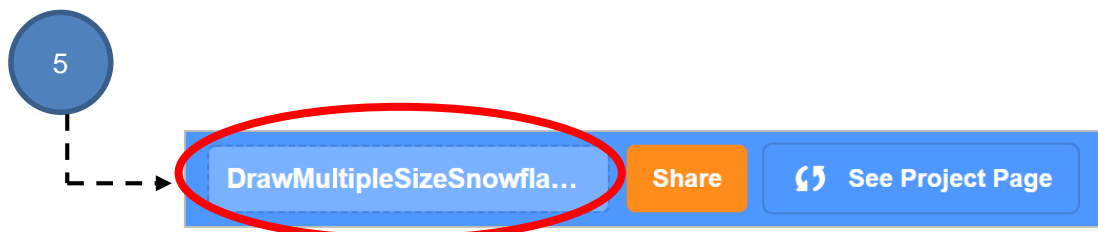
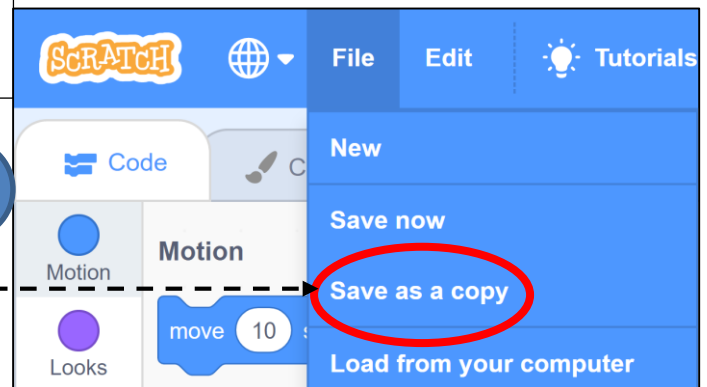
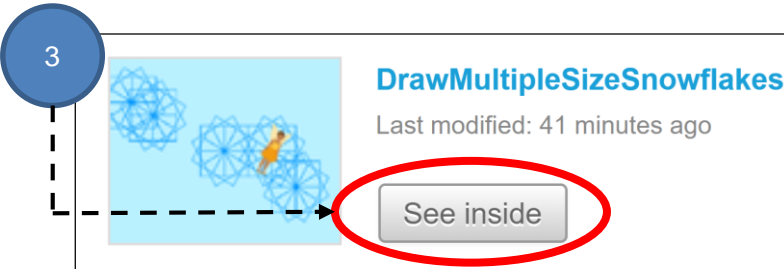


## START HERE

- ❑ Sign into your account at [scratch.mit.edu](https://scratch.mit.edu). 1
- ❑ Go to **My Stuff** under your name at the right top of the screen. 2
- ❑ Click on the See inside button to open your **DrawMultipleSnowflakes** project. 3



- ❑ Select **Save as a copy** from the **File** menu. 4
- ❑ Change the name to "DrawMultipleSizeSnowflakes" and save your project. 5

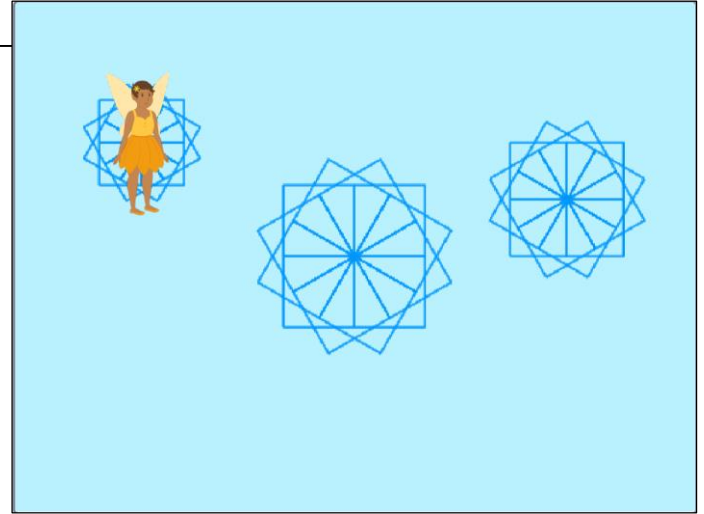


# COMPUTATIONAL ARTS WITH SCRATCH

## CHANGING THE SNOWFLAKE SIZE



Can you explain how to make snowflakes of different sizes at different locations?



- To make various sizes of snowflakes, we will need to change the size of the squares that make up the snowflakes.

What blocks should you use to change the snowflake size when you draw another snowflake?

Why don't you open your project and try it out? Once you finish, please share your findings with the class.



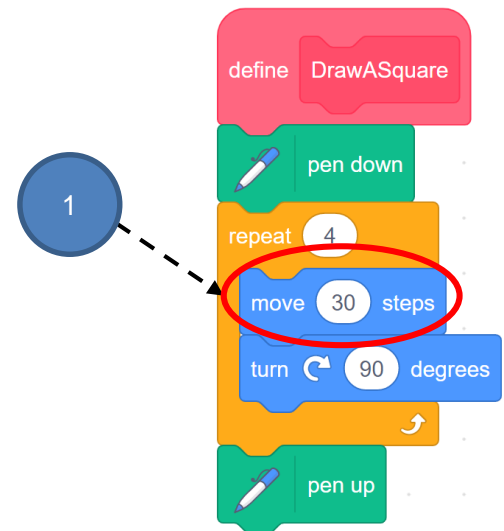
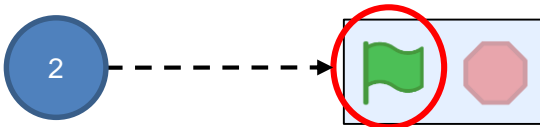
# COMPUTATIONAL ARTS WITH SCRATCH

## DRAWING DIFFERENT SIZED SNOWFLAKES

❑ To change the size of the snowflakes, you need to change the size of the squares.

1. Change the number of steps to **move 30 steps** in the **define DrawASquare** block.

2. Click on the **green flag** and see what is drawn.



❑ If you change the square size by changing the number of **move (x) steps** block, this will affect the sizes of all snowflakes.



How can you draw each snowflake in a different size?  
Do you have any ideas?

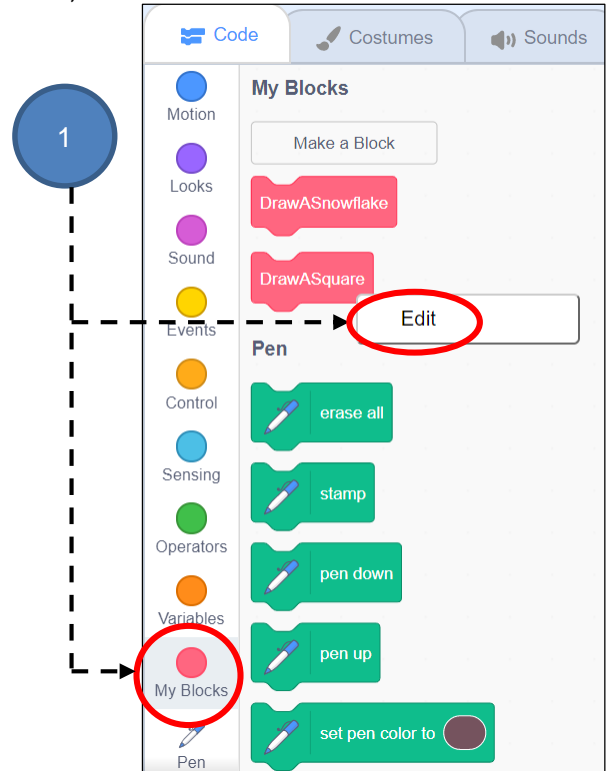
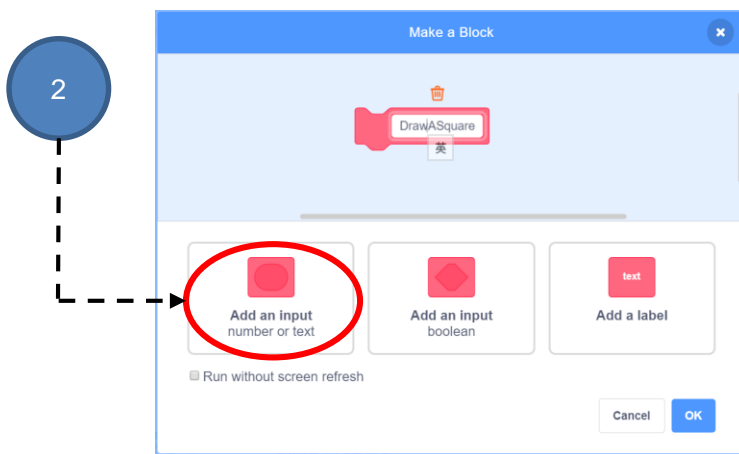
# COMPUTATIONAL ARTS WITH SCRATCH

You can create an input parameter for your custom block to create various sizes of squares and snowflakes.

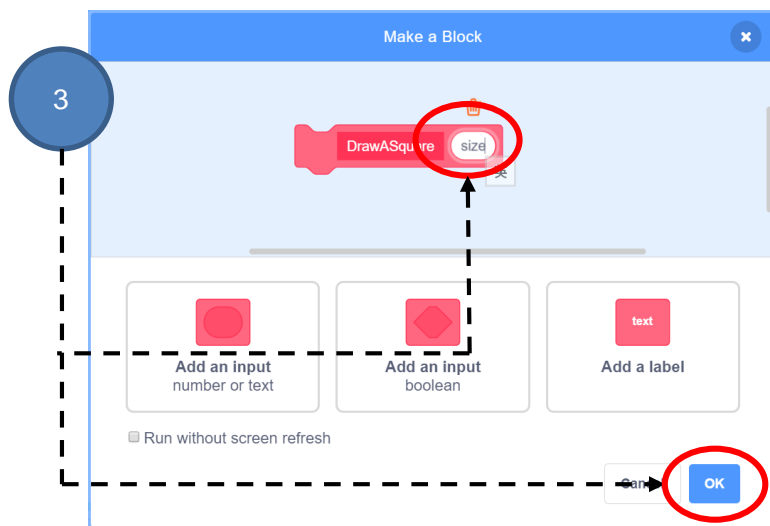
## DRAWASQUARE BLOCK

❑ To create an input parameter for your **DrawASquare** block, you need to:

1. Right click the **DrawASquare** block in the **My Blocks** drawer. Click on **Edit**.
2. Then click **Add an input**.



3. Type **size** to name the input parameter, then click the **OK** button to save the **DrawASquare** block.

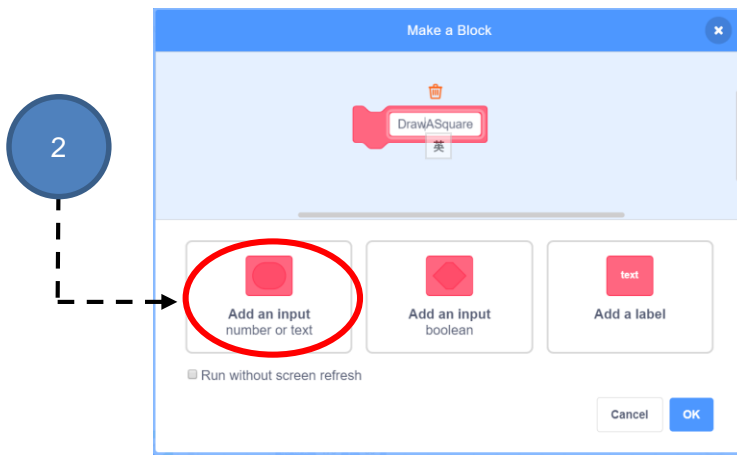


# COMPUTATIONAL ARTS WITH SCRATCH

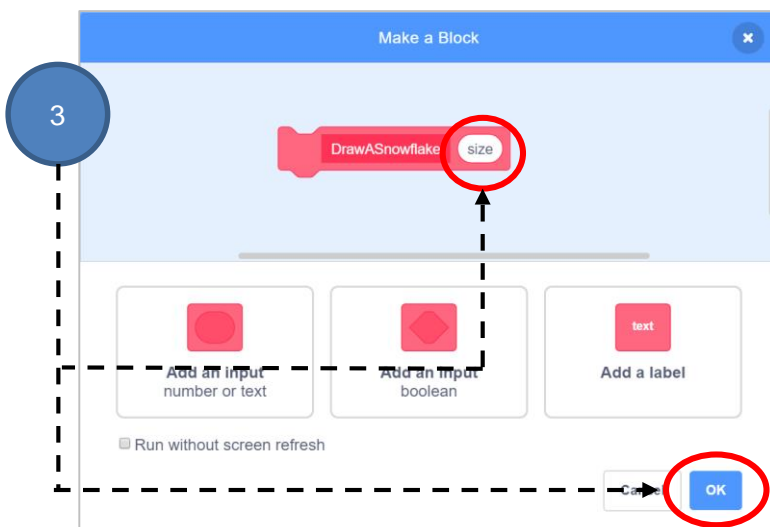
## DRAWASNOWFLAKE BLOCK

❑ Let's do the same with our **DrawASnowflake** block. Let's add an input parameter.

1. Right click the **DrawASnowflake** block in the **My Blocks** drawer. Click on **Edit**.
2. Then click **Add an input**.

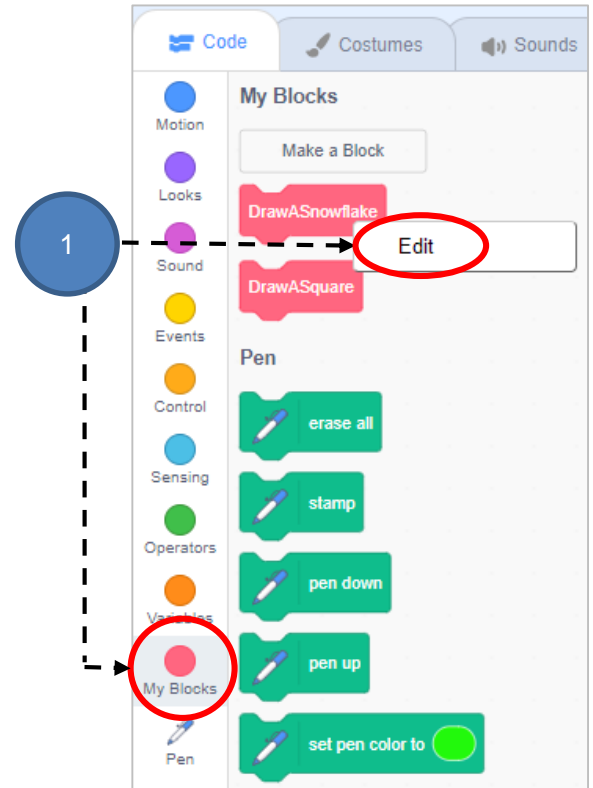


3. Type **size** to name the input parameter, then click the **OK** button to save the **DrawASnowflake** block.



**CT Tips**

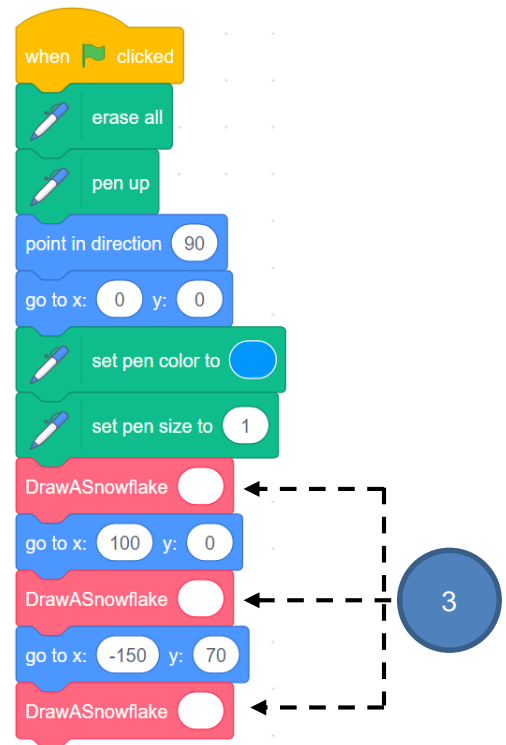
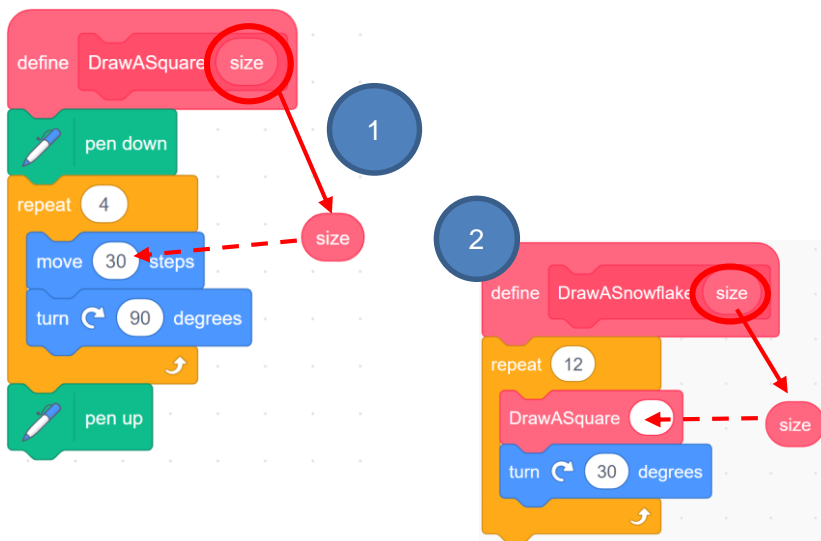
You can add an input parameter to your own blocks to make them more versatile, like varying the size of the squares and snowflakes.



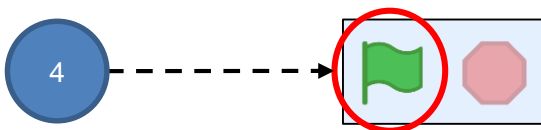
# COMPUTATIONAL ARTS WITH SCRATCH

## ADD INPUT PARAMETERS TO CUSTOM BLOCKS

- ❑ Set the size for **DrawASquare** and **DrawASnowflake** blocks.
  1. Drag **size** from **DrawASquare (size)** to replace **30** in the **move 30 steps** block in the **DrawASquare (size)** custom block.
  2. Drag **size** from **DrawASnowflake (size)** into the empty slot ( ) of the **DrawASquare ( )** block in the **DrawASnowflake (size)** custom block.
  3. Choose three different size values for each empty slot ( ) of the **DrawASnowflake ( )** blocks.



4. Click on the **green flag** and see what is drawn.



5. Now, can you add three more snowflakes to your project with different sizes and locations?

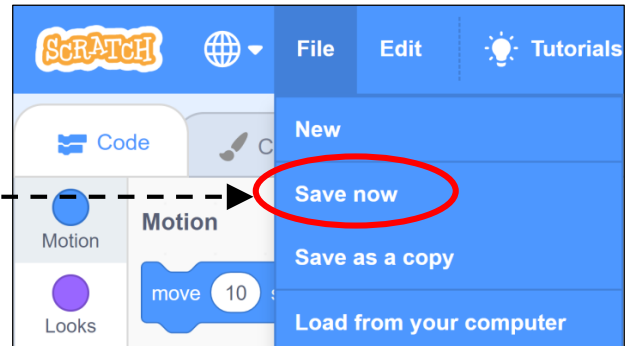
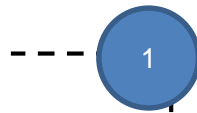
**CT Tips**  
An input parameter acts just like a variable. It can be used in place of a value in other blocks, like the move block.

# COMPUTATIONAL ARTS WITH SCRATCH

## SHARE YOUR PROJECT

When you finish, add your project to your teacher's studio:

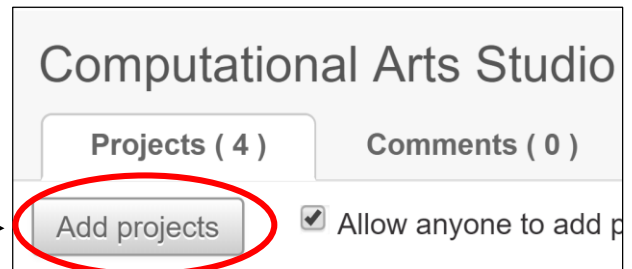
- Save your project by clicking **Save now** under the **File** menu.



- Click the orange **Share** button.



- Go to your teacher's Computational Arts Studio (they will give you a URL).



- Click the **Add projects** button.



- Select your project from the bottom of the page and add it to the studio.



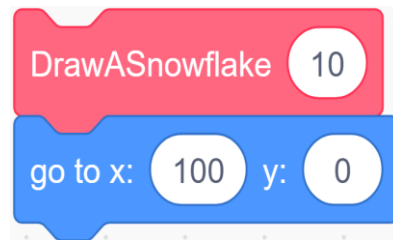
# COMPUTATIONAL ARTS WITH SCRATCH

## COMPUTATIONAL THINKING CONCEPTS

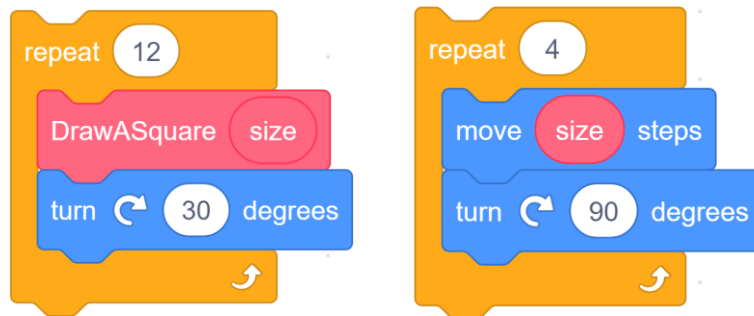
The following are the computational thinking concepts learnt in lesson 4.

### L1U8.7 - 8.8 Computational Arts with Scratch

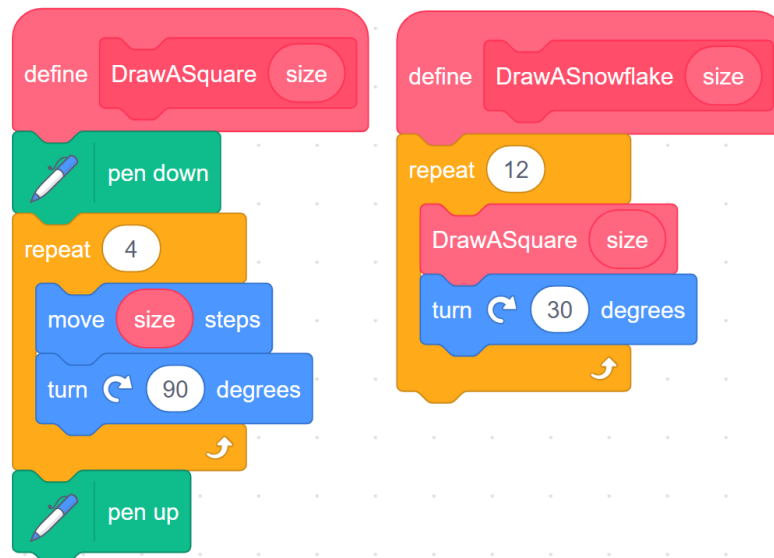
#### 1. Sequences:



#### 2. Repetition:



#### 3. Manipulation of data and elementary data structures:





# COMPUTATIONAL ARTS WITH SCRATCH

## COMPUTATIONAL THINKING PRACTICES

The following are the computational thinking practices used in this unit.

### L1U8.7 - 8.8 Computational Arts with Scratch

1. Reusing and remixing:

- a) Reuse and remix code blocks to draw squares and snowflakes from the subsequent lesson.

2. Being incremental and iterative:

- a) Start with a square, and iterate on that to draw a snowflake, eventually specifying different sizes.

3. Abstracting and modularizing:

- a) Create your own custom blocks to draw squares and snowflakes to make the code shorter and more versatile.

4. Testing and debugging:

- a) Test and debug to make sure the program draws a shape as expected.