

## Lab 3.2 Drawing Shapes (Again)

### Big Ideas

#### Products can be designed for life cycle

There are many advantages to creating custom blocks in SNAP, but one of them is that they will allow you to reuse code.

Imagine completing a program that uses a custom block to draw a smiley face. Anytime you require a smiley face in the program you can simply call the block of code. Later, you might work on a new project that requires a smiley face. If this is the case, you could easily reuse the block from your previous program.

This ability to easily reuse code, whether it be within one program, or within several programs, helps us understand the life cycle of products. It costs money to create and update programs. Anytime we can save money by speeding up the coding process means we can create a better product. It also means that as we create projects, we think further down the line about how this project might be maintained or altered later on. By using custom blocks of code, the maintenance and updates can be made easier.

In this lab, you will write code to draw regular polygons. But this time, you will write custom blocks and abstraction to write more efficient code.

### Simple Shapes

1. Write a SNAP script (or find one you've already written) to draw a square.
2. Write a SNAP script (or find one you've already written) to draw an equilateral triangle.
3. Write a SNAP script (or find one you've already written) to draw a regular pentagon.
4. Write a SNAP script (or find one you've already written) to draw a regular octagon.
5. Look over the four programs from above. Do you notice sections that are very similar? What sections might be able to be abstracted into a separate block?



## If You've Seen One...

1. Take your script from above that draws a square and turn it into a custom block.
2. Modify your custom block to use a variable for the number of sides instead of the number 4. Set that variable's value to be 4 so that you still draw a square.
3. Now, see if you can modify your block so that it will work correctly for any number of sides greater than 2. Look closely at the angles in your four scripts section 1 and see if you can spot a pattern.
4. Use your custom block in a program that asks the user for a number and then draws a regular polygon with that many sides. If the number given is less than 3, give an error message and do not draw anything.

## Bonus: Sizing Things Up

1. Modify your custom block and program from the previous section so that the user can specify both the number of sides and the size of each side. Be sure to utilize generalization and detail removal to make your program and block as clear and concise as possible.

### Curricular Competencies – Applied Design Understanding Context

As you allow for user input into the program, think carefully about the user and about they will interact with your program. What is the best way to prompt them for information? What is the best way for them to enter information? What is the best way to acknowledge their input, or to indicate they have inputted invalid data?

These questions need to be carefully considered as you design software for end users. Sometimes it's a good idea to interview end users before designing the software. That way you learn about their needs and the way they use software. It might also be a good idea to do some initial testing with end users so that you can see how they interact with the program and they can provide feedback in order to improve the software.

<b>Lab 3.2 Criteria</b>	
1 Custom block draws square	0.25 points
2 Use variable for number of sides	0.5 points
3 Works for all sides > 2	0.5 points
4a Program asks for sides, calls block	0.5 points
4b Program handles input <= 2	0.5 points
Bonus Program and block handle size of sides	0.5 points
Code clear and concise	0.25 points
<b>PROJECT TOTAL</b>	<b>3.0 points</b>