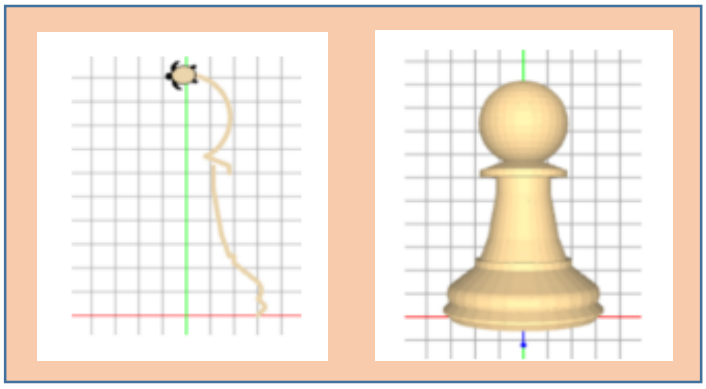
nclab
More Than Coding

TINA 2 PROGRAMMING COURSE

STUDENT JOURNAL

REVISED APRIL 2, 2016



NAME	
DATE STARTED	DATE COMPLETED
SCHOOL, CLASS, PERIOD	

TABLE OF CONTENTS:

General Website: https://nclab.com/

Turtle Gallery : https://nclab.com/turtle-gallery/

Desktop (needs login information) https://desktop.nclab.com/

Keep your name and password in a safe place.

LIST OF BASIC COMMANDS FROM TINA 1

`tina.go(n)`, where *n* = the number of steps.

This command moves the turtle forward, creating a line on the graph

It can also be written as

`tina.forward(n)`

`tina.fd(n)`

`tina.left(n)`, where *n* = the number of degrees to turn.

`tina.right(n)`, where *n* = the number of degrees to turn.

These commands will turn the turtle by the number of degrees indicated.

Left can also be written as `tina.lt(n)`, and right as `tina.rt(n)`

`tina.pu();tina.pd()`

Pen up (so that Tina moves forward without drawing a line) and pen down (to resume drawing with forward movement):

`tina.width(n)`, sets the line width, where *n* is the line width in units.

`tina.hide()` hides the turtle so that it doesn't show in the final drawing

`tina.back(d)`, where *d* is the distance Tina backs up. Tina will not draw when backing up.

`tina.extrude(n)`, where *n* is the width to be extruded. Extrude sets the width of the shape in the z axis so that the design can be printed. Extrude also hides Tina, so the hide command does not need to be used if the extrude command is used.

`tina.goto(x,y)`. Tina will draw a line to a specified coordinate pair. This is a useful command for angles that are not integers or decimals to the nearest tenth. It should only be used if necessary.

`tina.angle(n)`. Reset Tina to face a certain angle relative to the x axis.

LIST OF KEY VOCABULARY FROM TINA 1

Loop: Loop: a set of repeated commands. There are two different types of loops in Python: The counting **(for) loop** which repeats the set of commands a given number of times, and the conditional **(while) loop** which repeats the set of commands while a given condition is satisfied (the number of repetitions does not have to be known in advance). Tina only uses the for-loop. The set of commands that will be repeated is called the "body of the loop".

Algorithm: a series of logical steps that leads to the solution of a task. Students may be familiar with algorithms used in operations such as subtraction and long division.

Logical error: a mistake in an algorithm. Planning helps reduce the number of errors.

Computer Program: An algorithm written using a programming language.

Syntax: the way a command line is written.

Syntax error: a mistake in spelling, operators, indentations, spaces

Nested loops: a nested loop is a repeated pattern or loop that is part of another repeated pattern, or loop. For example: if I want to make a row of 10 triangles, I can write a loop for to make a triangle, and nest it inside a loop that will draw that triangle ten times in a row.

Variable: in terms of programming, variable is the name and value of something that will be recorded in memory. In the For loop, the i is an **index or counting variable**. If we set i to a range of values, then i will change each time the loop starts the body of the program. If we then use i as part of a command, that command will output a different value each time.

Range: the range is the lower and upper limit of the variable i. Note that if the range is set with only one value, then the lower limit of the range is assumed to be 0, with the number in the parentheses being the upper limit. Important: the final value will be the difference between the upper and lower limit. For example, in range (1,11), the final value used in the program is 10, or 11-1.

SECTION 6: PARAMETRIC VARIABLES

In Section 6, learn to use parametric variables in a program.

Parametric (adjective), parameter (noun): is made up of the prefix para-, which means "beside", and metron, which means "measure". Both are Greek in origin. A **parameter** is a numerical factor that sets the conditions for the operation of a system.

In the program, we can write a statement for a parametric variable **before** the program itself. In the program, we used the name of the variable instead of the actual value. We can change the value of the parametric variable without rewriting the program.

Make a list of the parametric variables used in Section 6:

LEVEL	PARAMETRIC VARIABLES	CONDITIONS CONTROLLED BY THE VARIABLE
6.1 Swiss Cross II		
6.2 Beehive		
6.3 Windmill II		
6.4 Snowflake II		
6.5 Polygon		
6.6 Mosaic IV		
6.7 Gear II		

QUESTIONS TO THINK ABOUT WHILE YOU GO THROUGH SECTION 6

Use examples from the levels in Section 6 to help you answer the questions.

What advantages are there to writing parametric variables?
How could you use parametric variables to test designs in art, crafts, trades, engineering, or science? Think of a couple of different examples.
When do you use variables, and when do you just assign a fixed value?

SECTION 6: PARAMETRIC VARIABLE ART PROJECT

DESIGNING THE ART PROJECT

Draw the design.

Describe the patterns and shapes. How do they change? What functions and commands will you need to draw them? How will you name the function? Describe the colors, starting positions, line lengths, turning angles, and coordinates for the `goto` command if needed.

Write out the order in which you will program each set of commands (functions, loops, main program). Try modeling the program as a flow chart.

REFLECTION ON PARAMETRIC VARIABLE ART PROJECT

How did you organize the program (dividing tasks into functions and main program)?

How many functions did you use?

What would you like to create with Tina?

Art Project File Name: _____

File is located at _____

Screen Shot of Art Project

NOTES ON SECTION 6: PARAMETRIC VARIABLES

SECTION 7: FUNCTIONS

In Section 7, learn how to write functions, which are sets of commands written separately from the main program.

`Def`: the `def` command is used to define a function.

Insert words from the list into the explanation for the following function:

`def` polygon (T,e,n):

means

function	number
edge	define
turtle	

"Define a _____ named polygon for _____T, length of _____ e, _____of sides n."

What variables are used in this function? Explain what each one does in the function. Notice that these are different than counting variables used in a For statement.

HAVE FUN WITH THE FUNCTIONS IN EACH LEVEL!

ONCE YOU HAVE SOLVED THE PROBLEM, TRY ASSIGNING DIFFERENT VALUES IN THE MAIN PROGRAM.

SEE WHAT THE FUNCTION DOES WHEN YOU RUN THE PROGRAM.

QUESTIONS TO THINK ABOUT WHILE YOU GO THROUGH SECTION 7

Use examples from the levels in Section 6 to help you answer the questions.

How are functions and parametric variables similar?
How are functions and parametric variables different?
How could you use functions to test designs in art, crafts, trades, engineering, or science? Think of a couple examples, different from those you chose for Section 6.

SECTION 7: FUNCTION ART PROJECT

DESIGNING THE ART PROJECT

Draw the design.

Describe the patterns and shapes. How do they change? What functions and commands will you need to draw them? How will you name the function? Describe the colors, starting positions, line lengths, turning angles, and coordinates for the `goto` command if needed.

Write out the order in which you will program each set of commands (functions, loops, main program). Try modeling the program as a flow chart.

REFLECTION ON FUNCTION ART PROJECT

How did you organize the program (dividing tasks into functions and main program)?

How many functions did you use?

What would you like to create with Tina?

Art Project File Name: _____

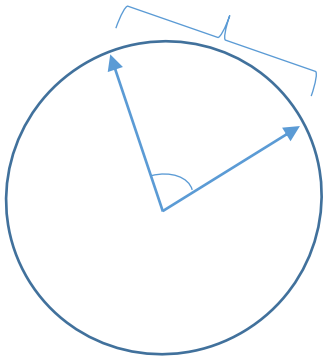
File is located at _____

Screen Shot of Art Project

NOTES ON SECTION 7: FUNCTIONS

SECTION 8: ARCS

In Section 8, learn how to use the `arc()` command to create curved lines.

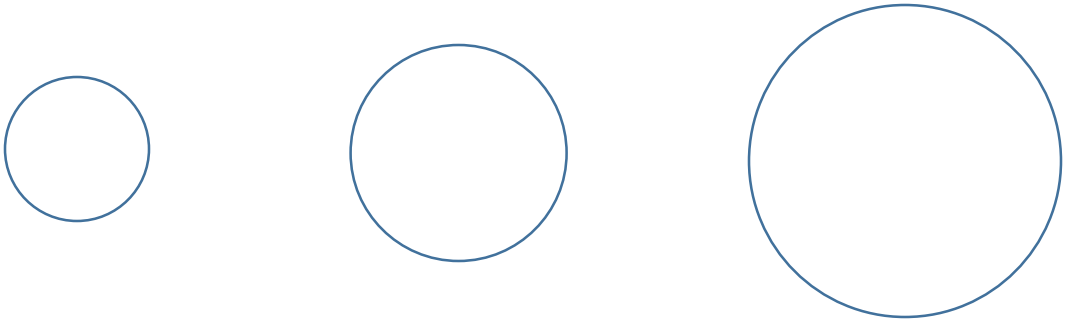


Label the **radius**, **angle** and **arc** in the diagram.

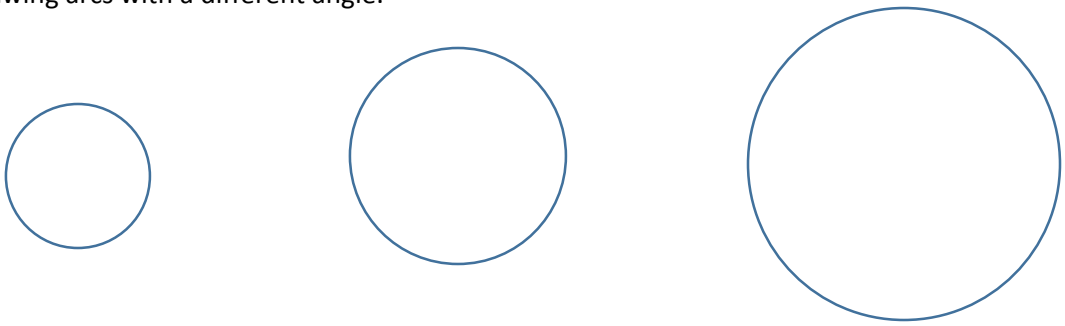
The length (size) and shape of the arc are determined by the radius and angle.
The command is written as `tina.arc(a, b)`, where *a* is the number of degrees, and *b* is the length of the radius.

A third parameter can be included, telling Tina which direction to turn when writing the arc:
`tina.arc(a, b, 'r')` tells Tina to turn right. Note the single quotation marks.
`tina.arc(a, b, 'l')` tells Tina to turn left

Practice drawing arcs using circles with different radii and the same angle. How does this change the appearance of the arc?



Now try drawing arcs with a different angle.



QUESTIONS TO THINK ABOUT WHILE YOU GO THROUGH SECTION 8

What three components are needed to describe an arc?
Can you visualize a portion of a circumference? How do you decide what values to use?
Can arcs as used in Section 8 be used to draw any curve?

SECTION 8: INITIALS ART PROJECT

DESIGNING THE ART PROJECT

Draw the design.

Describe the patterns and shapes of the letters. What arcs will you need to draw them? Describe the colors, starting positions, line lengths, turning angles, and coordinates for the `goto` command if needed.

Write out the order in which you will program the letters. When will the up, down, and back commands be useful?

REFLECTION ON INITIALS ART PROJECT

How many lines of programming did your project need to create your initials?

How did you organize the program (dividing tasks into functions and main program)?

How many arcs did you use? Did you run into any problems designing the arcs?

Which parts of the design process were difficult? Which parts were easy?

What would you like to design next?

Art Project File Name: _____

File is located at _____

Screen Shot of Art Project

NOTES ON SECTION 8: ARCS

SECTION 9: SOLIDS

In Section 9, learn how to create rotational solids using the command `rosol()`.

The rotational solid is created simply by rotating a trace around the y axis. However, writing a program that just lists commands (go, left, right, etc.) will be limited to only one shape. We want to use functions to allow us to vary the resulting shape.

Make notes as you go through the levels. How are functions used to make the program more flexible and powerful?

Since the shapes are rotated about an axis, think of the distances from the y-axis as radii.

Level 9.1
Level 9.2
Level 9.3
Level 9.4
Level 9.5
Level 9.6
Level 9.7

QUESTIONS TO THINK ABOUT WHILE YOU GO THROUGH SECTION 9

Review the steps needed to create a 3D solid.
What are some limitations to using `rosol()`?
What objects could you model using rotational solids?

SECTION 9: SOLIDS

DESIGNING THE ART PROJECT

Draw the design.

Describe the design. What commands will you need to draw the traces? Describe the colors, starting positions, line lengths, turning angles, and coordinates for the `goto` command if needed.

Write out the order in which you will program the components? Is it better to start from the bottom or the top of the shapes? When will the up, down, back, and angle commands be useful?

REFLECTION ON ROTATED SOLID ART PROJECT

How did you use variables in your design?

Which parts of the design process were difficult? Which parts were easy?

What would you like to design next?

Art Project File Name: _____

File is located at _____

Screen Shot of Art Project

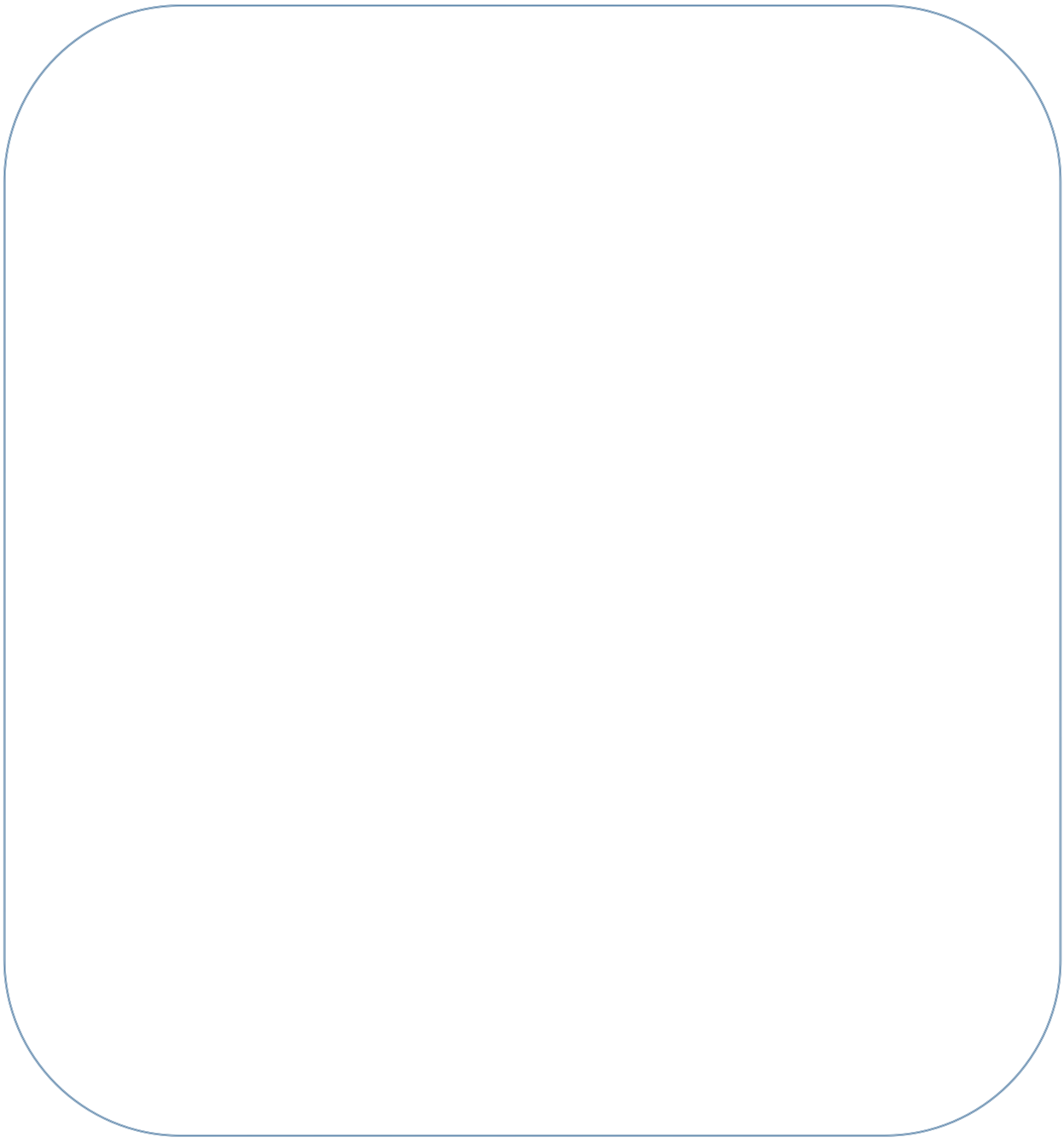
NOTES ON SECTION 9: SOLIDS

SECTION 10: SURFACES AND SHELLS

In Section 10, learn how to draw rotational surfaces and shells using `rosurf()` and `roshell()`.

Time to make another collection!

Collect photographs or drawings of hollow shapes based on cylinders and cones in art, crafts, nature, manufacturing, engineering, and science. Paste them in these boxes.



When do you use `rosurf()`?
When do you use `roshell()`?

SECTION 10: ROTATIONAL SHELL ART PROJECT

DESIGNING THE ART PROJECT

Draw the design.

Describe the patterns and shapes. How will you use functions and variables? Describe the colors, starting positions, line lengths, turning angles, and coordinates for the `goto` command if needed.

Write out the order in which you will program the traces. Will you start at the bottom or the top? When will the up, down, back, and angle commands be useful?

REFLECTION ON ROTATIONAL SHELL ART PROJECT

How did you use functions and variables in your design?

If you were able to print the design, how did it turn out? Would you make changes to your design next time?

Which parts of the design process were difficult? Which parts were easy?

What would you like to design next? Are you ready to learn more 3D design techniques? What would you like to learn?

Art Project File Name: _____

File is located at _____

Screen Shot of Art Project

NOTES ON SECTION 10: SURFACES AND SHELLS

DESIGNER LOG - DESIGNS I HAVE CREATED

File Name	Date created	Description

The following grid page can be used for your designs. Make a clear plastic overlay to use on the grid, or make copies for sketching and designing when you don't have a computer handy.

NOTES