# Session 4: Polygons and Curves

## Housekeeping

Make separate functions for:

1. Writing the starting part of the gcode
2. Writing the ending part of the gcode
3. Making each individual shape and adding it directly to the gcode file.

This way, we may not need the structure variable as we did in the previous session.

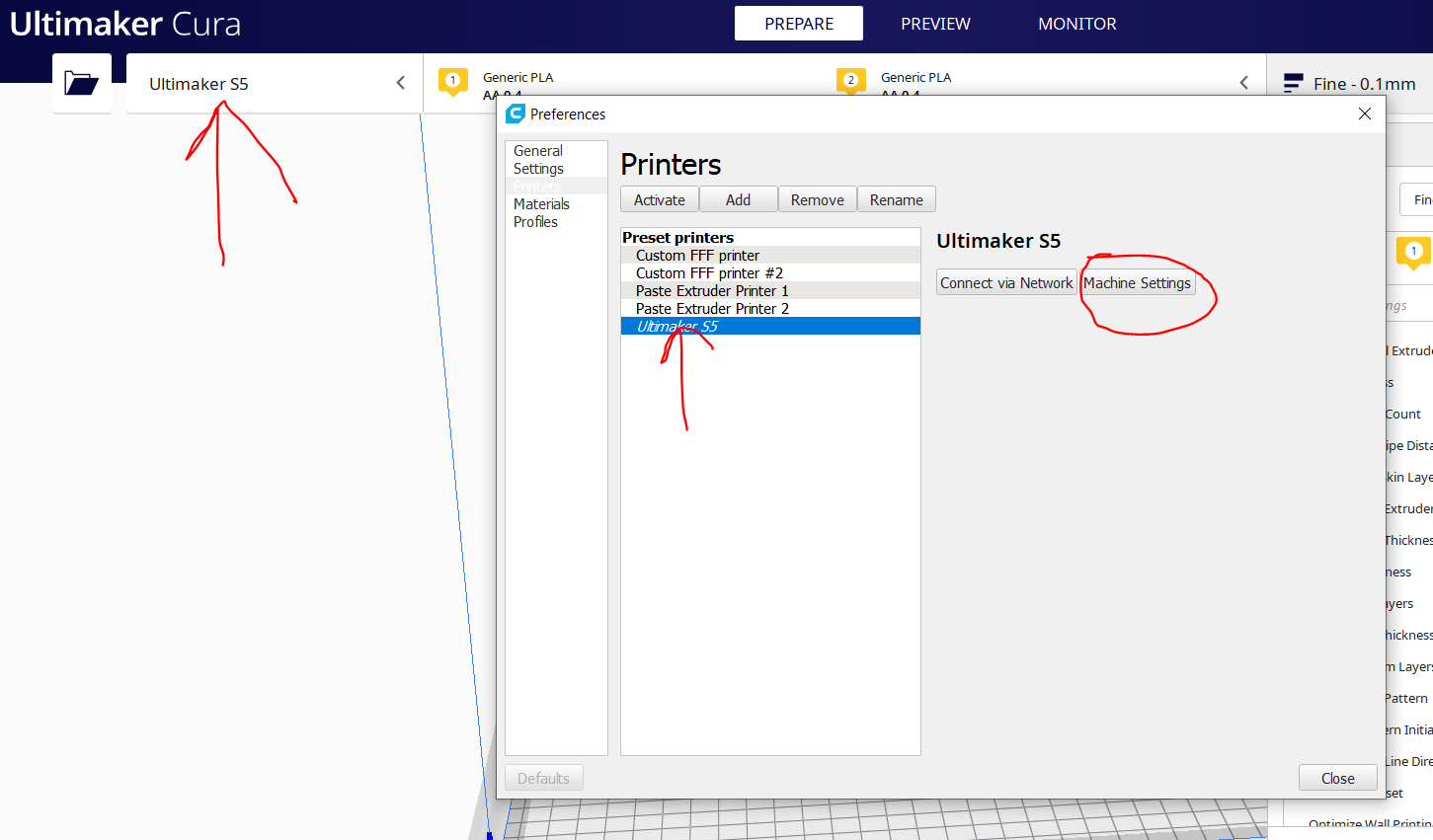
Also, make a separate file for functions and include that in the main function using the #include line. This way, we don’t have to make one cpp file too big and can keep adding more features as needed.

Reference Videos for including:

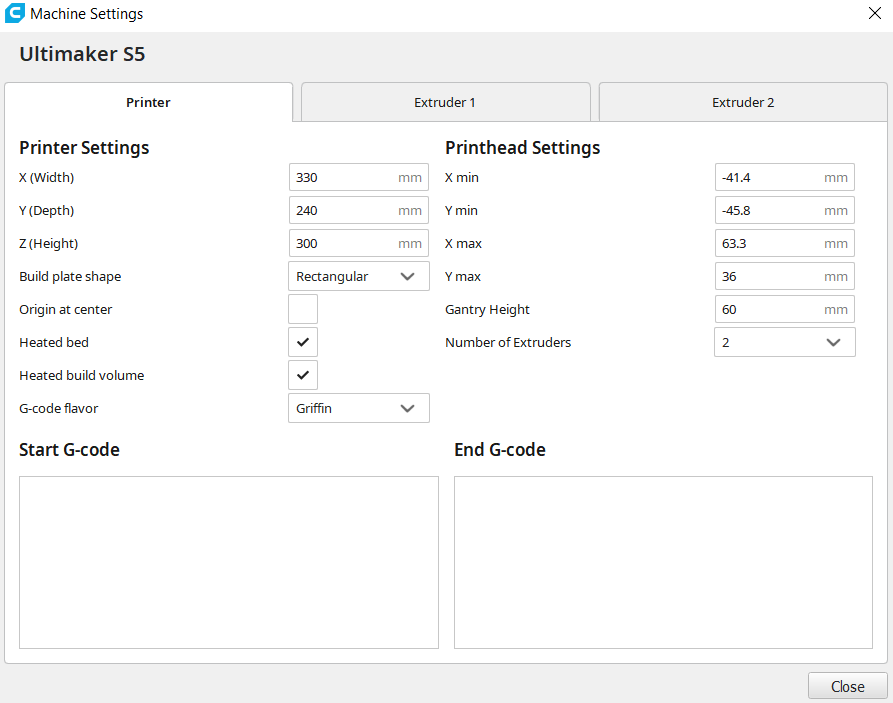
1. Making header files and including them with .h extension: <https://www.youtube.com/watch?v=9RJTQmK0YPI>
2. Header files in general: <https://www.youtube.com/watch?v=qaGzc56Rekg>

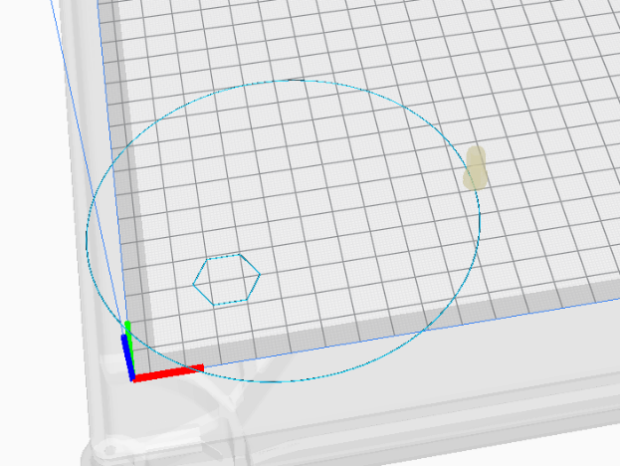
Ensure that you add the z raising from the current layer in every function during travel.

Also include a function to check if the thing being printed is within the print boundary. To find print boundary go to the top bar and click on the printer name and find machine settings.

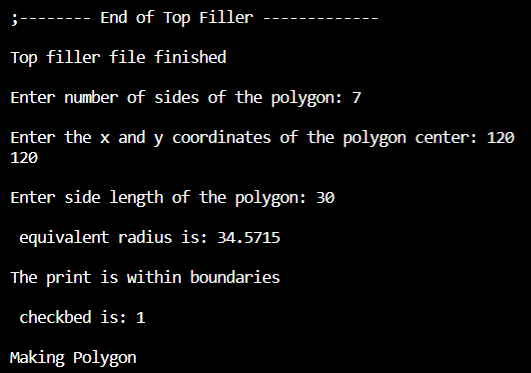


This shows the size of the x(width) and y(depth) which is the maximum values for the movement in x and y directions.



So, the boundaries for Ultimaker S5 printer are:

1. X (10,320) since bed size is 330
2. Y (10,230) since bed size is 240

What a problem of a circle going outside the bed would look like this where material is just hanging in the air.

When properly avoided, the user interaction should show that the print is inside the permissible area.

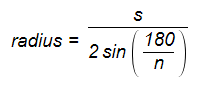
## Function to make Polygons

To make a polygon, we must reach every vertex of it in a loop. This would work for all regular polygons the same way.

Algorithm to make a polygon:

1. Find radius of the circumcircle
2. Locate all the points on the circumference
3. Plot a path going through all of them

### Radius of the circumcircle

The reference is [here](https://www.mathopenref.com/polygonradius.html#:~:text=Radius%20of%20a%20regular%20polygon%20(also%20Circumradius)&text=The%20radius%20of%20a%20regular,that%20passes%20through%20every%20vertex.).

The radius is given by the formula:

Where,

s: is the length of any side

n: is the number of sides

sin: is the sine function calculated in degrees

However, C++ calculates sine in radians. So, we change this to pi/n instead of 180/n.

### Location of the points

The image here shows a unit circle and the coordinates of all the points of this pentagon.

This formula can be generalized for any polygon as:

and

Where x0,y0 is the center of the polygon’s circumcenter and j is the vertex number starting from horizontal.

Assignment: Write a function to make a polygon anywhere on the bed and with any number of sides.

## Function to make circle

In order to make a circle, we just make a polygon with a 100 or more sides depending on the size of the circle and it approximates a circle.

Here, we don’t know the side length so we must use a distance calculator function to get the distance between 2 adjacent points.

Use the same distance formula between current point and new point and multiply that distance with er to get how much extrusion is to be used.

Use the circle and polygon functions one after the other to make concentric shapes or offset shapes. Try to ensure that the shapes don’t cross since we are still printing within the same plane.

This image is of a pentagon of side length 10 and a circle of radius 5 at the same center.

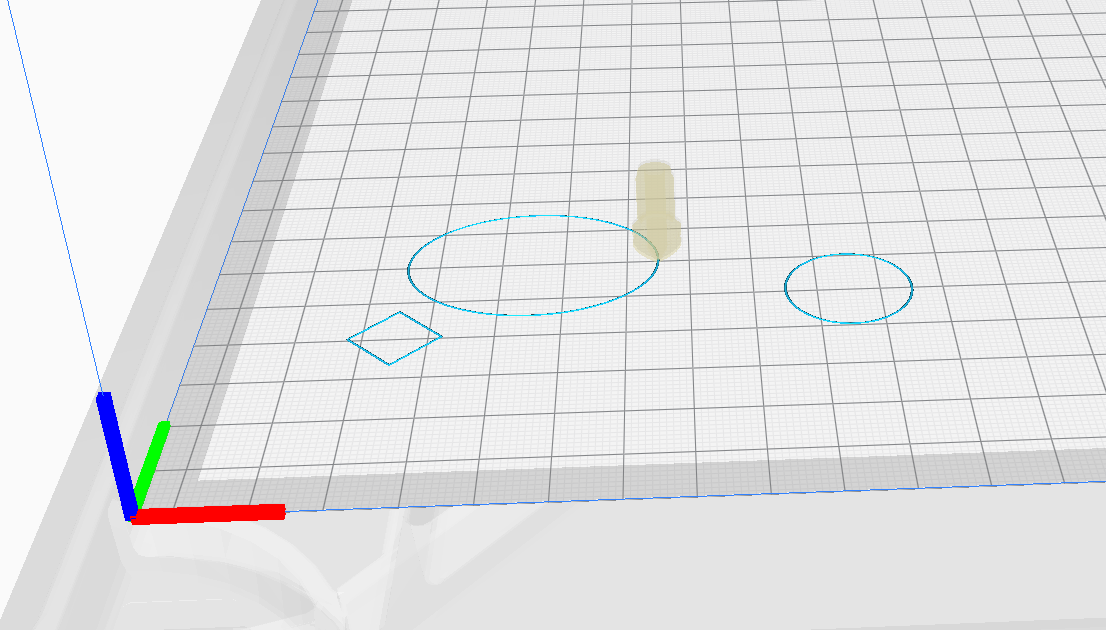
## Function to make ellipse

The same principle applies to an ellipse as a circle. Except, here there are 2 radii to use, the major and minor axes.

The equation of the ellipse is:

We can write and

Here’s a website explaining this equation: <https://www.mathopenref.com/coordparamellipse.html>

And that is how we get the x and y coordinates of every point in the ellipse. To make the whole ellipse, just like the circle, we go from theta = 0 to 2\*pi.

Try to make a code where all of the shapes are printed one after the other by asking the user for the dimensions and positions of all of the items.

Additional things you can try:

1. Think of a way to avoid overlapping shapes.
2. Try to make an inclined ellipse. (major axis not horizontal).
3. Try to make polygons where the start angle is not zero.

Assignment: Write a function for making circles and another function to make ellipses.

## Function to fill in shapes

Filling in a shape is just like printing concentric shapes until the whole inner area is filled. At the center, there will be some space left as for a given thickness of extrusion, the next concentric shape cannot fit inside. Thus, we need to purge some extra material in the center to fill up the gap.

Until now we were extruding based on length or distance between 2 points. However, now we need to fill an area. Given the thickness of extrusion, the amount of material to be filled in the center would be:

Where,

D is the equivalent extrusion distance to be used in

R is the effective radius of the smallest shape that can be made

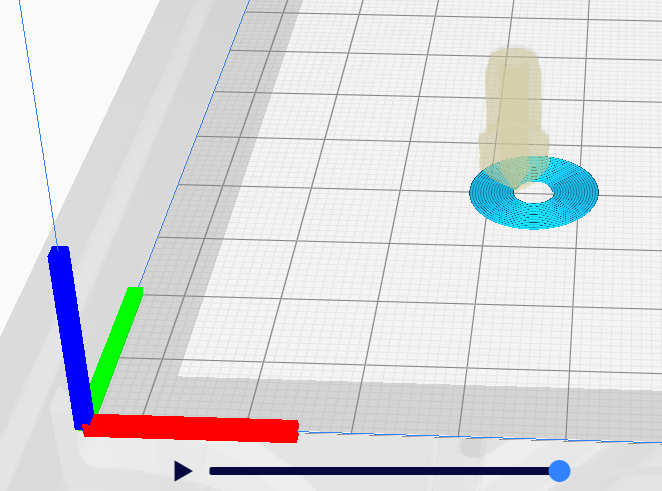
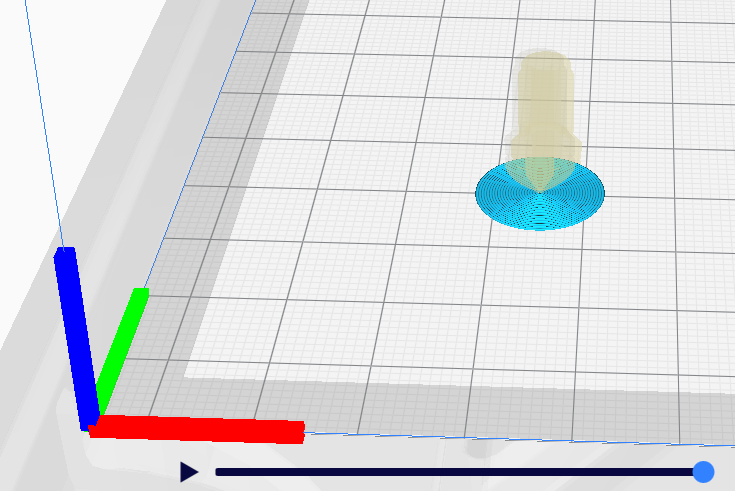
A and B are the major and minor axes of the ellipse

W is the extrusion thickness (width)

This area/ length gives an approx. estimate of the distance to be used.

Instead of having separate functions that take on all the variables, we just input the values from the user in the function itself.

### Filled circle



Here, the thickness used is 0.3mm. Given that the z axis height by default is 0.2mm, we should use W = 0.2mm for an actual print to ensure all the strands have good contact and make a solid object.

### Filled Ellipse

Since the center gap is longer here unlike the circle, we use a filler line-segment instead of the usual dot. Note that you need to change the direction for the line if the major axis is smaller than the minor axis. (if a<b then the filler line at the end would be vertical instead of horizontal).

Think about how much material needs to be extruded here. It is not merely covering the length since there is a bulge in between. Best to use the area of the remaining ellipse and divide by the width to get the effective distance of extrusion.

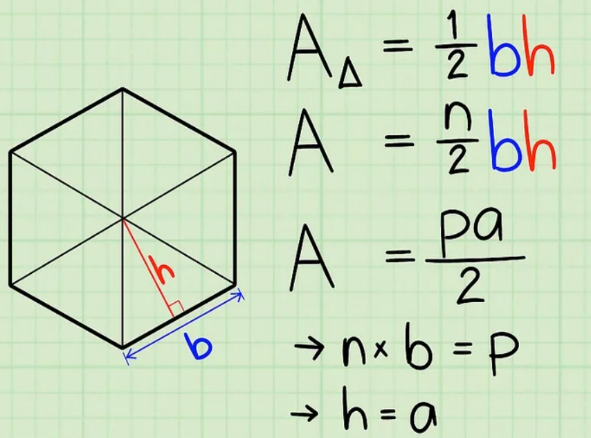
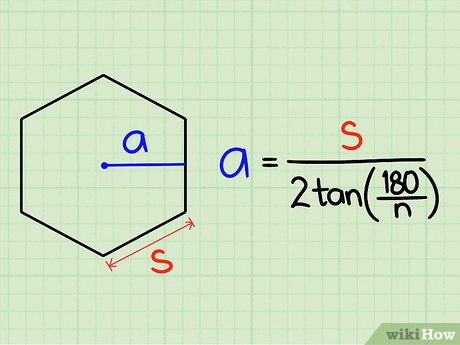
### Filled Polygons

The gap in the center again needs to be filled with an equivalent of the area divided by the print width.

The equation for area of a polygon is:

Where n is the number of sides.

The derivation is given in this site: <https://www.wikihow.com/Find-the-Area-of-Regular-Polygons>



Assignment: Write one program that chooses which shape you want to make out of polygon, circle or ellipse and the corresponding function makes that filled shape for you.

Additional things you can try:

1. Make a shape where the outside is filled instead of the inside. This can be a hole.
2. Make a filled shape of anything other than a polygon, circle or ellipse.