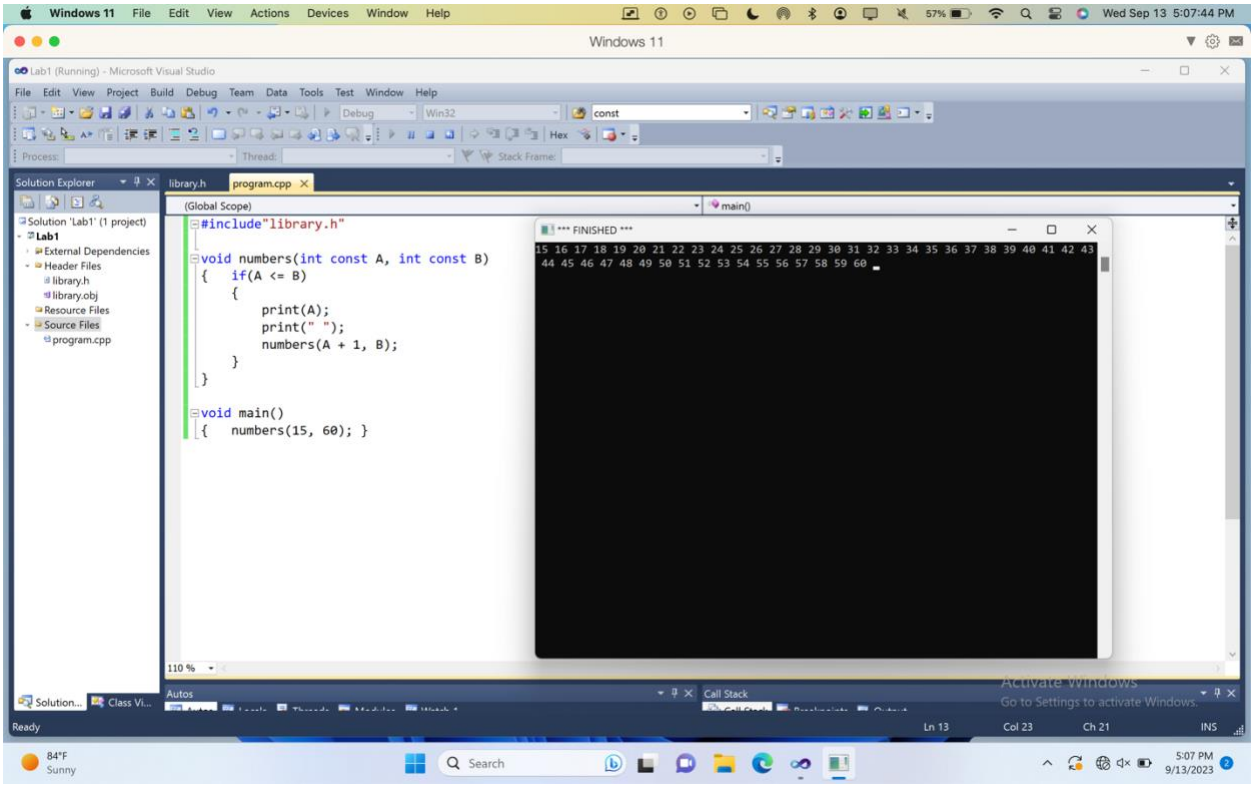
Lab 3

Sloan Atkins

ECE 118

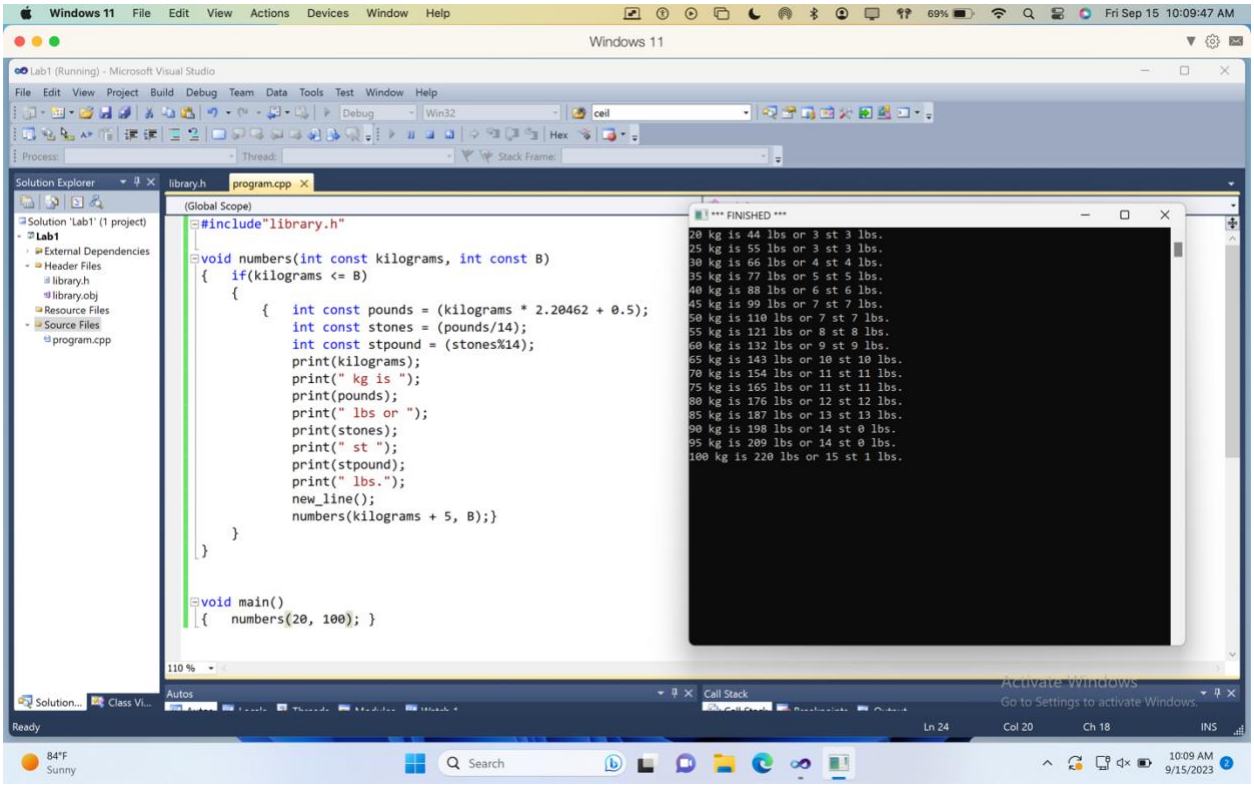
A1. *Remembering how to start*

```
#include"library.h"

void numbers(int const A, int const B)
{	if(A <= B)
	{
		print(A);
		print(" ");
		numbers(A + 1, B);
	}
}

void main()
{	numbers(15, 60); }
```



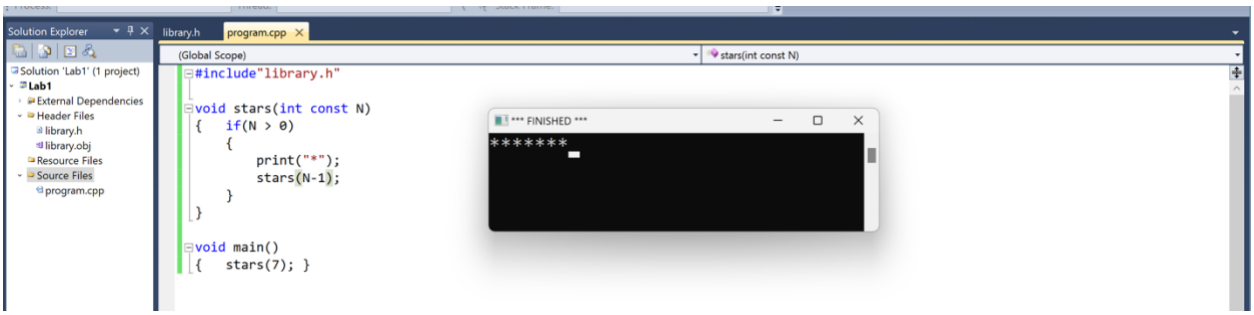
A2. *Exotic Measures*

```
#include"library.h"

void numbers(int const kilograms, int const B)
{	if(kilograms <= B)
	{
		{	int const pounds = (kilograms * 2.20462 + 0.5);
			int const stones = (pounds/14);
			int const stpound = (stones%14);
			print(kilograms);
			print(" kg is ");
			print(pounds);
			print(" lbs or ");
			print(stones);
			print(" st ");
			print(stpound);
			print(" lbs.");
			new_line();
			numbers(kilograms + 5, B);}
	}
}


void main()
{	numbers(20, 100); }
```



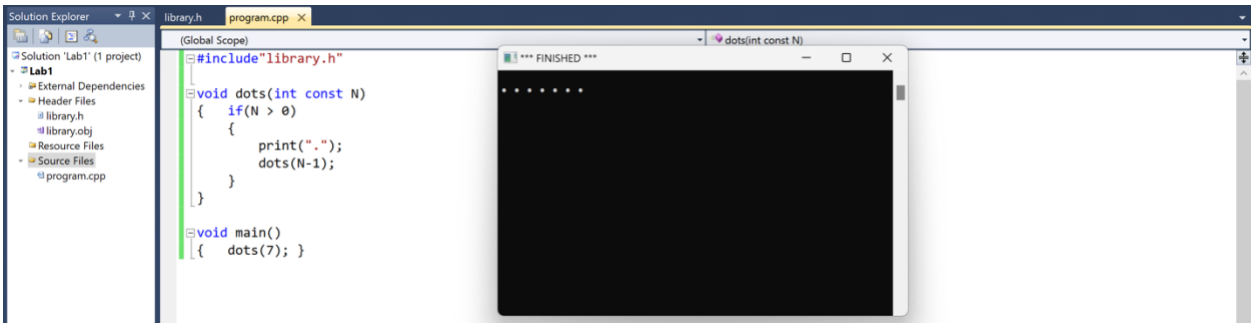
B1. *Stars*

```
#include"library.h"

void stars(int const N)
{	if(N > 0)
	{
		print("*");
		stars(N-1);
	}
}

void main()
{	stars(7); }
```



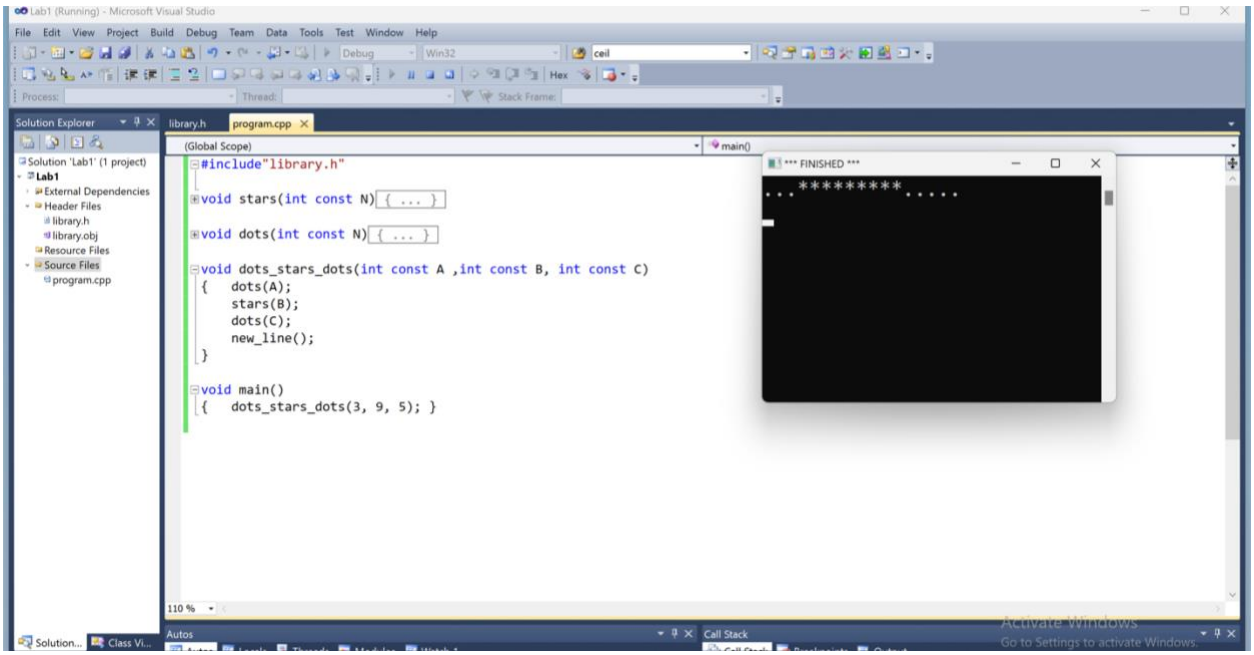

B2. *Dots*

```
#include"library.h"
void dots(int const N)
{	if(N > 0)
	{
		print(".");
		dots(N-1);
	}
}

void main()
{	dots(7); }
```



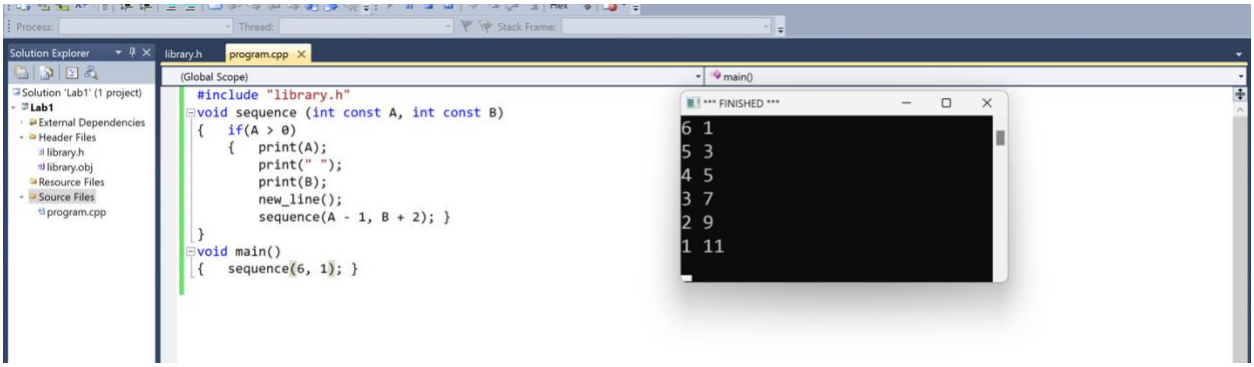
B3. *Stars and Dots*

```
#include"library.h"

void stars(int const N)
{	if(N > 0)
	{
		print("*");
		stars(N-1);
	}
}

void dots(int const N)
{	if(N > 0)
	{
		print(".");
		dots(N-1);
	}
}

void dots_stars_dots(int const A ,int const B, int const C)
{	dots(A);
	stars(B);
	dots(C);
	new_line();
}

void main()
{	dots_stars_dots(3, 9, 5); }
```



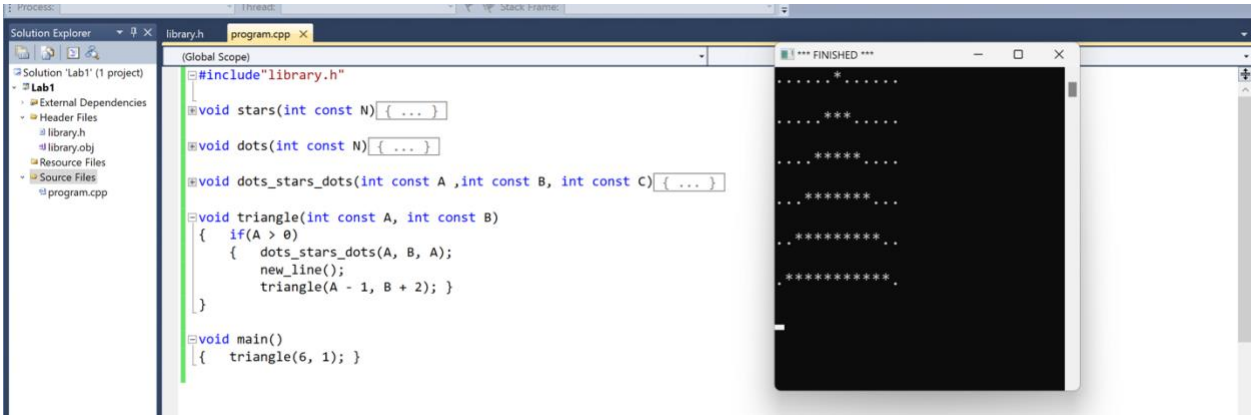
B4. *Another adaptation*

```
#include "library.h"
void sequence (int const A, int const B)
{     if(A > 0)
      {     print(A);
            print(" ");
            print(B);
            new_line();
            sequence(A - 1, B + 2); }
}
void main()
{     sequence(6, 1); }
```



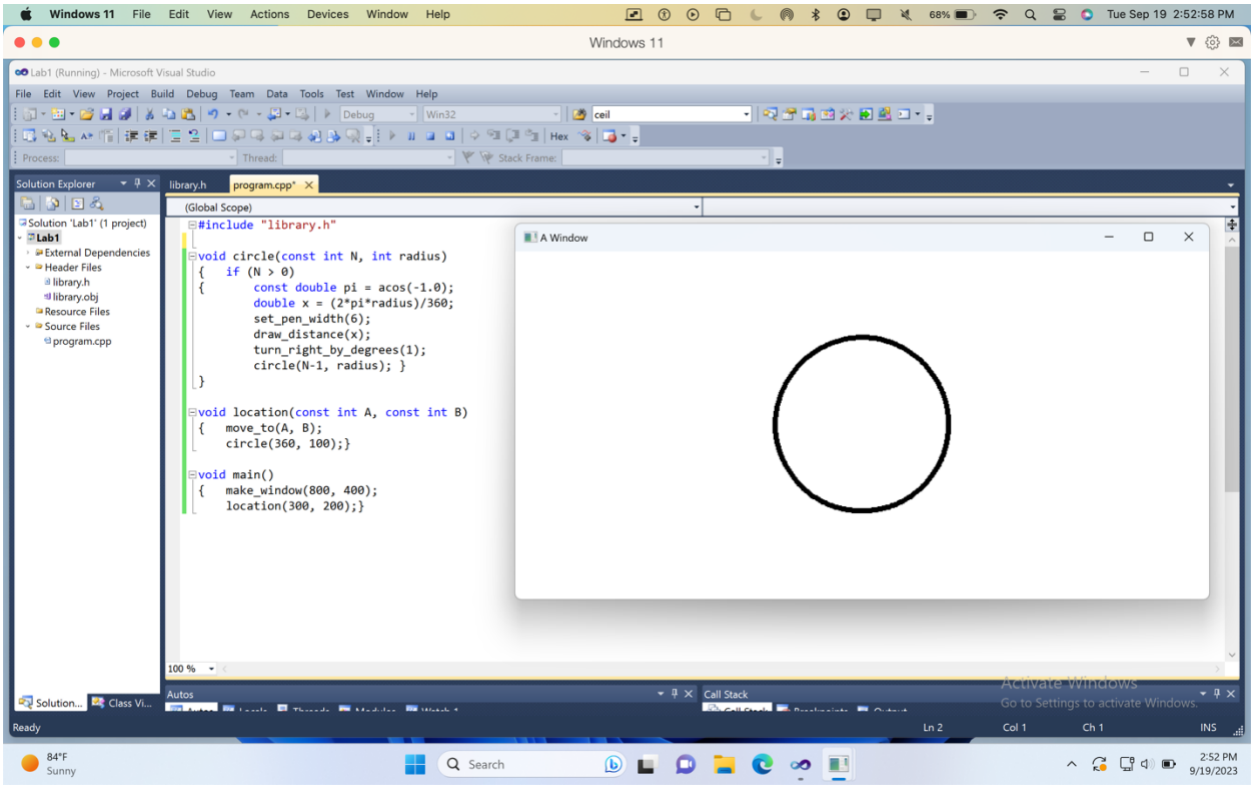
B5. *Combining*

```
#include"library.h"

void stars(int const N)
{	if(N > 0)
	{
		print("*");
		stars(N-1);
	}
}

void dots(int const N)
{	if(N > 0)
	{
		print(".");
		dots(N-1);
	}
}

void dots_stars_dots(int const A ,int const B, int const C)
{	dots(A);
	stars(B);
	dots(C);
	new_line();
}

void triangle(int const A, int const B)
{	if(A > 0)
	{	dots_stars_dots(A, B, A);
		new_line();
		triangle(A - 1, B + 2); }
}

void main()
{	triangle(6, 1); }
```



C1. *A circle*

```
#include "library.h"

void circle(const int N, int radius)
{      if (N > 0)
{             const double pi = acos(-1.0);
              double x = (2*pi*radius)/360;
              set_pen_width(6);
              draw_distance(x);
              turn_right_by_degrees(1);
              circle(N-1, radius); }
}

void location(const int A, const int B)
{      move_to(A, B);
       circle(360, 100);}

void main()
{      make_window(800, 400);
       location(300, 200);}
```



C2. *Weaponizing*

```
#include "library.h"
void circle(const int N, int radius, int degrees)
{      if (N > 90 - degrees)
{             const double pi = acos(-1.0);
              double x = (2*pi*radius)/360;
              set_pen_width(6);
              draw_distance(x);
              turn_right_by_degrees(1);
              circle(N-1, radius, degrees);}
}

void draw_cannon(const double radius)
{      set_pen_color(color::black);
       set_pen_width(5);
       start_shape();
       turn_left_by_degrees(90);
       draw_distance(0.5*radius);
       turn_right_by_degrees(90);
       draw_distance(1.25*radius);
       turn_right_by_degrees(94);
       draw_distance(4*radius);
       turn_right_by_degrees(87);
       draw_distance(0.85*radius);
       turn_right_by_degrees(87);
       draw_distance(2.68*radius);
}

void location(const int A, const int B)
{      move_to(A, B);
       circle(720, 40, 50);
       draw_cannon(100);}

void main()
{      make_window(800, 400);
       location(300, 300);}
```