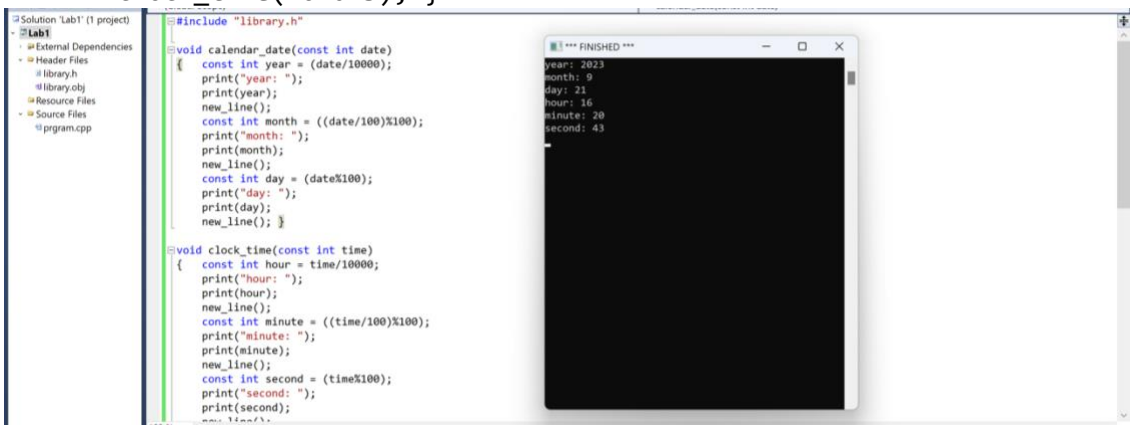
Lab #5

ECE 118 – Section R/RC
Lab on Wednesday at 5:05
Sloan Atkins

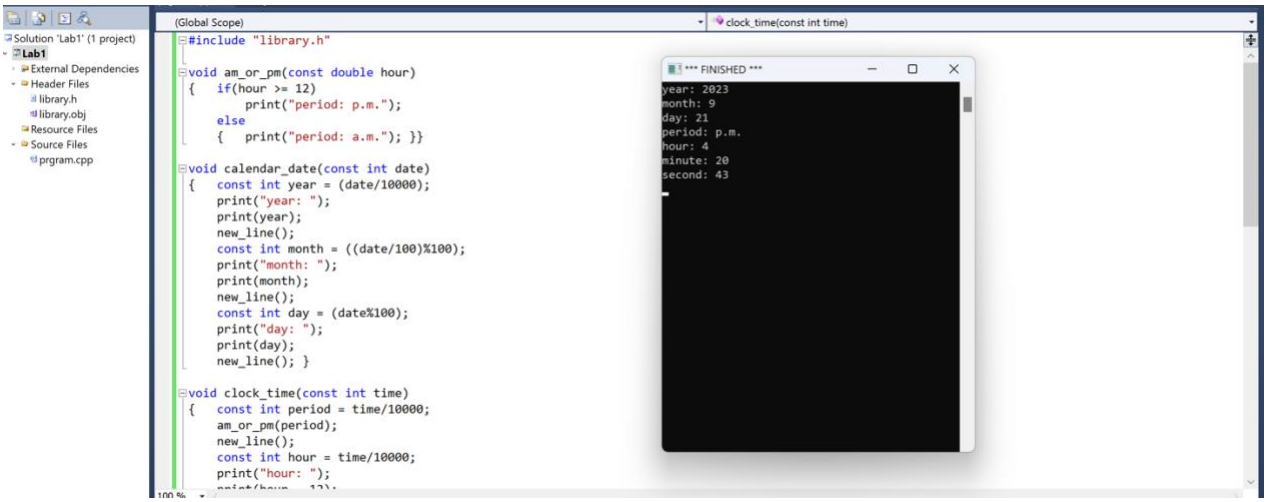
1. What Time Is It?

```
#include "library.h"

void calendar_date(const int date)
{	const int year = (date/10000);
	print("year: ");
	print(year);
	new_line();
	const int month = ((date/100)%100);
	print("month: ");
	print(month);
	new_line();
	const int day = (date%100);
	print("day: ");
	print(day);
	new_line(); }

void clock_time(const int time)
{	const int hour = time/10000;
	print("hour: ");
	print(hour);
	new_line();
	const int minute = ((time/100)%100);
	print("minute: ");
	print(minute);
	new_line();
	const int second = (time%100);
	print("second: ");
	print(second);
	new_line();
	print(""); }

void main()
{	calendar_date(20230921);
	clock_time(162043); }
```



2. The Monroe Doctrine.

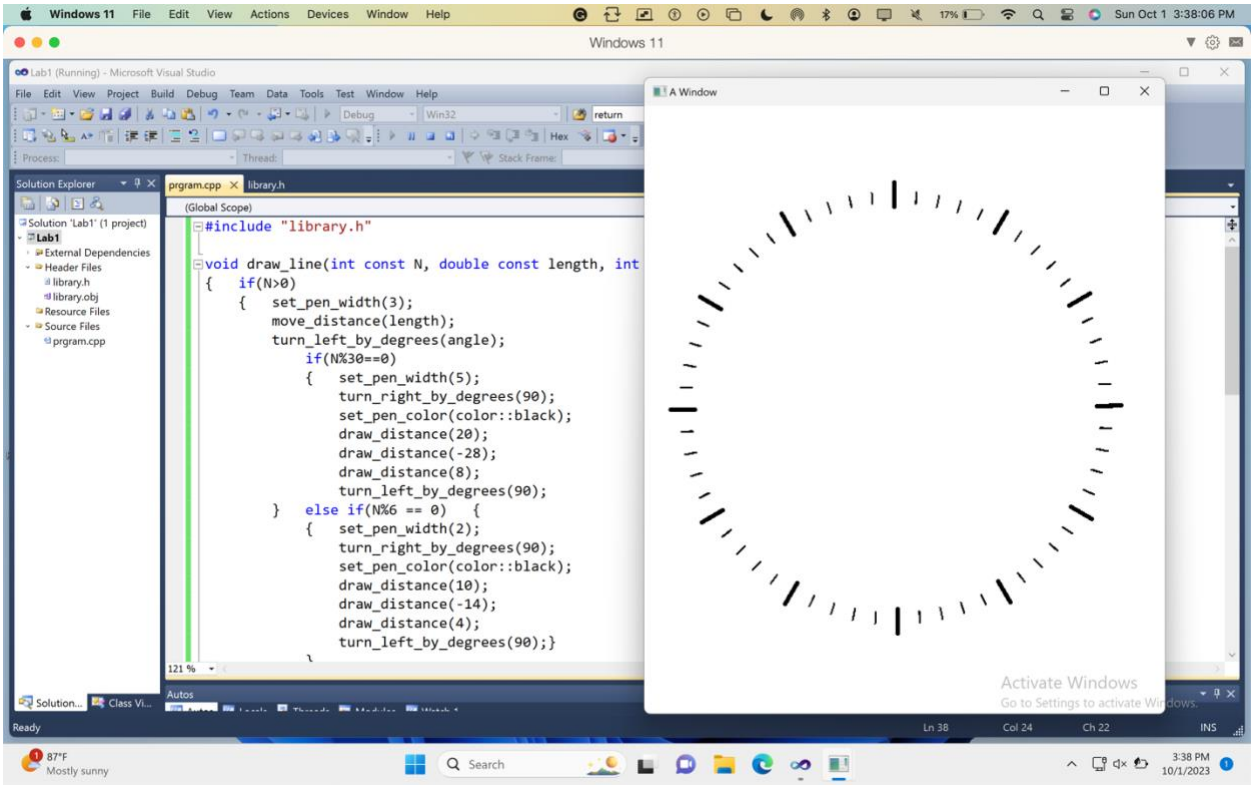
```
#include "library.h"

void am_or_pm(const double hour)
{     if(hour >= 12)
            print("period: p.m.");
      else
      {     print("period: a.m."); }}

void calendar_date(const int date)
{     const int year = (date/10000);
      print("year: ");
      print(year);
      new_line();
      const int month = ((date/100)%100);
      print("month: ");
      print(month);
      new_line();
      const int day = (date%100);
      print("day: ");
      print(day);
      new_line(); }

void clock_time(const int time)
{     const int period = time/10000;
      am_or_pm(period);
      new_line();
      const int hour = time/10000;
      print("hour: ");
      print(hour - 12);
      new_line();
      const int minute = ((time/100)%100);
      print("minute: ");
      print(minute);
      new_line();
      const int second = (time%100);
      print("second: ");
      print(second);
      new_line();
      print(""); }

void main()
{     calendar_date(20230921);
      clock_time(162043); }
```



3. A Clock Face.

```
#include "library.h"

void draw_line(int const N, double const length, int const angle)
{	if(N>0)
	{	set_pen_width(3);
		move_distance(length);
		turn_left_by_degrees(angle);
			if(N%30==0)
			{	set_pen_width(5);
				turn_right_by_degrees(90);
				set_pen_color(color::black);
				draw_distance(20);
				draw_distance(-28);
				draw_distance(8);
				turn_left_by_degrees(90);
		}	else if(N%6 == 0)  {
			{	set_pen_width(2);
				turn_right_by_degrees(90);
				set_pen_color(color::black);
				draw_distance(10);
				draw_distance(-14);
				draw_distance(4);
				turn_left_by_degrees(90);}
			}
			draw_line(N-1, length, angle);
	}
}
```

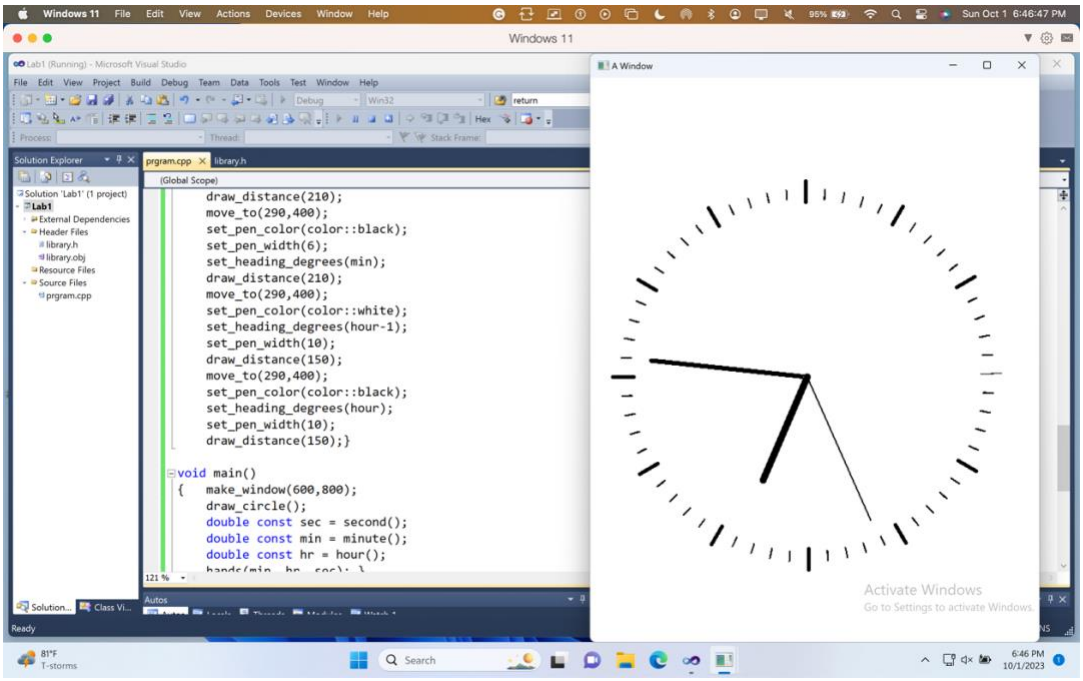
```
void draw_circle()
{      int const x=250;
       int const y=350;
       int const radius=240;
       move_to (x+radius+40,y);
       double const length = 2*radius*acos(-1.0)/360;
       draw_line(360,length,1);}

void main()
{      make_window(600,700);
       draw_circle();}
```



4. A Whole Clock

```
#include "library.h"

double hour()
{	int const time = get_clock_time();
	double const hour = time/10000;
	double const min = (time/100)%100;
		if(hour<=12)
		{	double const hourangle = (hour/12+min/720)*360;
			return hourangle;}
		else if(hour>12)
		{	double const hourangle = ((hour-12)/12+min/720)*360;
			return hourangle;}
		return 0;}

double minute()
{	int const time = get_clock_time();
	double const minute = (time/100)%100;
	double const minuteangle= minute/60.0*360;
	return minuteangle;}

double second()
{	int const time = get_clock_time();
	double const second = time%100;
	double const secondangle = second/60.0*360;
	return secondangle;}

void draw_line(int const N, double const length, int const angle)
{	if(N>0)
	{	set_pen_width(3);
		move_distance(length);
		turn_left_by_degrees(angle);
			if(N%30==0)
			{	set_pen_width(5);
				turn_right_by_degrees(90);
				set_pen_color(color::black);
				draw_distance(20);
				draw_distance(-28);
				draw_distance(8);
				turn_left_by_degrees(90);
		}	else if(N%6 == 0)  {
			{	set_pen_width(2);
				turn_right_by_degrees(90);
				set_pen_color(color::black);
				draw_distance(10);
				draw_distance(-14);
				draw_distance(4);
				turn_left_by_degrees(90);}
			}
			draw_line(N-1, length, angle);
```

```
	}
}

void draw_circle()
{	int const x=250;
	int const y=350;
	int const radius=240;
	move_to (x+radius+40,y+50);
	double const length = 2*radius*acos(-1.0)/360;
	draw_line(360,length,1);}

void hands(double const min,double const hour, const int sec)
{	move_to(290,400);
	set_pen_color(color::white);
	set_pen_width(2);
	set_heading_degrees(sec-6.0);
	draw_distance(210);
	move_to(290,400);
	set_pen_color(color::black);
	set_pen_width(2);
	set_heading_degrees(sec);
	draw_distance(210);
	move_to(290,400);
	set_pen_color(color::white);
	set_pen_width(6);
	set_heading_degrees(min-6.0);
	draw_distance(210);
	move_to(290,400);
	set_pen_color(color::black);
	set_pen_width(6);
	set_heading_degrees(min);
	draw_distance(210);
	move_to(290,400);
	set_pen_color(color::white);
	set_heading_degrees(hour-1);
	set_pen_width(10);
	draw_distance(150);
	move_to(290,400);
	set_pen_color(color::black);
	set_heading_degrees(hour);
	set_pen_width(10);
	draw_distance(150);}

void main()
{	make_window(600,800);
	draw_circle();
	double const sec = second();
	double const min = minute();
	double const hr = hour();
	hands(min, hr, sec); }
```

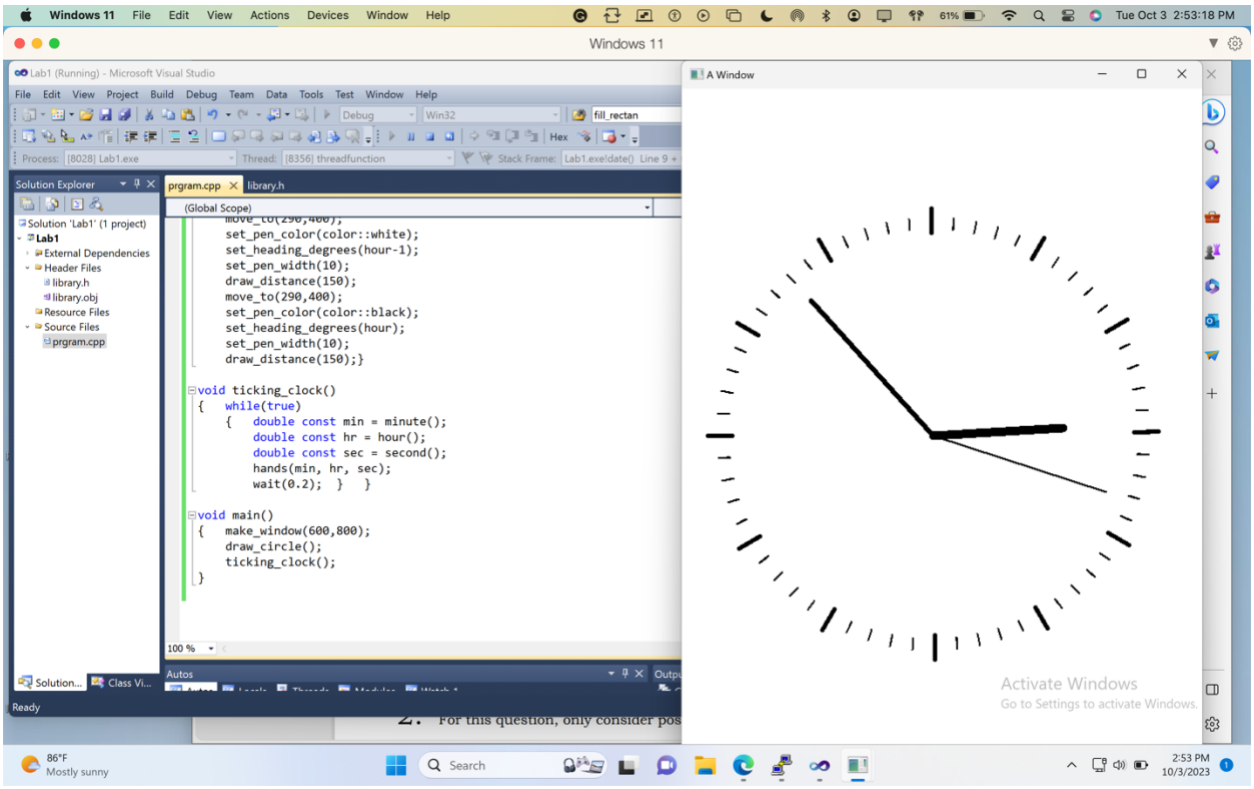

Windows 11
Lab1 (Running) - Microsoft Visual Studio
A Window
draw_distance(210);
move_to(290,400);
set_pen_color(color::black);
set_pen_width(6);
set_heading_degrees(min);
draw_distance(210);
move_to(290,400);
set_pen_color(color::white);
set_heading_degrees(hour-1);
set_pen_width(10);
draw_distance(150);
move_to(290,400);
set_pen_color(color::black);
set_heading_degrees(hour);
set_pen_width(10);
draw_distance(150);}
void main()
{ make_window(600,800);
draw_circle();
double const sec = second();
double const min = minute();
double const hr = hour();
Activate Windows
Go to Settings to activate Windows.
Ready
Search
6:46 PM
10/1/2023

5. Animate Your Clock.

```
#include "library.h"

double hour()
{	int const time = get_clock_time();
	double const hour = time/10000;
	double const min = (time/100)%100;
		if(hour<=12)
		{	double const hourangle = (hour/12+min/720)*360;
			return hourangle;}
		else if(hour>12)
		{	double const hourangle = ((hour-12)/12+min/720)*360;
			return hourangle;}
		return 0;}

double minute()
{	int const time = get_clock_time();
	double const minute = (time/100)%100;
	double const minuteangle= minute/60.0*360;
	return minuteangle; }

double second()
{	int const time = get_clock_time();
	double const second = time%100;
	double const secondangle = second/60.0*360;
	return secondangle; }

void draw_line(int const N, double const length, int const angle)
{	if(N>0)
	{	set_pen_width(3);
		move_distance(length);
		turn_left_by_degrees(angle);
			if(N%30==0)
			{	set_pen_width(5);
				turn_right_by_degrees(90);
				set_pen_color(color::black);
				draw_distance(20);
				draw_distance(-28);
				draw_distance(8);
				turn_left_by_degrees(90);
		}	else if(N%6 == 0)  {
			{	set_pen_width(2);
				turn_right_by_degrees(90);
				set_pen_color(color::black);
				draw_distance(10);
				draw_distance(-14);
				draw_distance(4);
				turn_left_by_degrees(90);}
			}
			draw_line(N-1, length, angle);
	}
```

```
}

void draw_circle()
{     int const x=250;
      int const y=350;
      int const radius=240;
      move_to (x+radius+40,y+50);
      double const length = 2*radius*acos(-1.0)/360;
      draw_line(360,length,1);}

void hands(double const min,double const hour, const int sec)
{     move_to(290,400);
      set_pen_color(color::white);
      set_pen_width(2);
      set_heading_degrees(sec-6.0);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::black);
      set_pen_width(2);
      set_heading_degrees(sec);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::white);
      set_pen_width(6);
      set_heading_degrees(min-6.0);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::black);
      set_pen_width(6);
      set_heading_degrees(min);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::white);
      set_heading_degrees(hour-1);
      set_pen_width(10);
      draw_distance(150);
      move_to(290,400);
      set_pen_color(color::black);
      set_heading_degrees(hour);
      set_pen_width(10);
      draw_distance(150);}

void ticking_clock()
{     while(true)
      {     double const min = minute();
            double const hr = hour();
            double const sec = second();
            hands(min, hr, sec);
            wait(0.2);   }     }

void main()
{     make_window(600,800);
```

```
        draw_circle();
        ticking_clock();
}
```



6. A Complete Product

```
#include "library.h"

void date()
{	int const date = get_calendar_date();
	int const year = date/10000 ;
	int const month = (date/100)%100;
	int const day = (date%100);
	set_pen_color(color::black);
	set_font("Ariel", 50);
	move_to(125,720);
	write_string(day);
		if(day==1 || day==21)
		{	write_string("st "); }
		if(day==2 || day==22)
		{	write_string("nd "); }
		if(day==3 || day==23)
		{	write_string("rd "); }
		else
		{	write_string("th "); }

		if(month==1)
		{	write_string("January "); }
		if(month==2)
		{	write_string("February "); }
		if(month==3)
		{	write_string("March "); }
		if(month==4)
		{	write_string("April "); }
		if(month==5)
		{	write_string("May "); }
		if(month==6)
		{	write_string("June "); }
		if(month==7)
		{	write_string("July "); }
		if(month==8)
		{	write_string("August "); }
		if(month==9)
		{	write_string("September "); }
		if(month==10)
		{	write_string("October "); }
		if(month==11)
		{	write_string("November "); }
		if(month==12)
		{	write_string("December "); }
		write_string(year);}

void time()
{	move_to(125,80);
	set_pen_color(color::black);
	set_font("Ariel", 80);
```

```
	int const time = get_clock_time();
	int const hour = time/10000;
	int const minute = (time/100)%100;
		if(hour<=12)
		{	write_string(hour);}
		else if(hour>12)
		{	write_string(hour-12);}

	write_string(":");
		if(minute<10)
		{	write_string("0");
			write_string(minute);}
		else
		{	write_string(minute);}

		if(hour<=12)
		{	write_string(" a.m.");}
		else if(hour>12)
		{	write_string(" p.m.");}
}

double hour()
{	int const time = get_clock_time();
	double const hour = time/10000;
	double const min = (time/100)%100;
		if(hour<=12)
		{	double const hourangle = (hour/12+min/720)*360;
			return hourangle; }
		else if(hour>12)
		{	double const hourangle = ((hour-12)/12+min/720)*360;
			return hourangle; }
		return 0;
}

double minute()
{	int const time = get_clock_time();
	double const minute = (time/100)%100;
	double const minuteangle= minute/60.0*360;
	return minuteangle; }

double second()
{	int const time = get_clock_time();
	double const second = time%100;
	double const secondangle = second/60.0*360;
	return secondangle; }

void draw_line(int const N, double const length, int const angle)
{	if(N>0)
	{	set_pen_width(3);
		move_distance(length);
		turn_left_by_degrees(angle);
			if(N%30 == 0)
```

```
                    {     set_pen_width(5);
                          turn_right_by_degrees(90);
                          set_pen_color(color::black);
                          draw_distance(20);
                          draw_distance(-28);
                          draw_distance(8);
                          turn_left_by_degrees(90); }
                    else if(N%6 == 0)
                    {     set_pen_width(2);
                          turn_right_by_degrees(90);
                          set_pen_color(color::black);
                          draw_distance(10);
                          draw_distance(-14);
                          draw_distance(4);
                          turn_left_by_degrees(90);}
                    draw_line(N-1, length, angle);
      }
}

void draw_circle()
{     int const x=250;
      int const y=350;
      int const radius=240;
      move_to (x+radius+40,y+50);
      double const length = 2*radius*acos(-1.0)/360;
      draw_line(360,length,1);}

void hands(double const sec,double const min, const int hour)
{     move_to(290,400);
      set_pen_color(color::white);
      set_pen_width(2);
      set_heading_degrees(sec-6.0);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::black);
      set_pen_width(2);
      set_heading_degrees(sec);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::white);
      set_pen_width(6);
      set_heading_degrees(min-6.0);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::black);
      set_pen_width(6);
      set_heading_degrees(min);
      draw_distance(210);
      move_to(290,400);
      set_pen_color(color::white);
      set_heading_degrees(hour-1);
      set_pen_width(10);
```

```
	draw_distance(150);
	move_to(290,400);
	set_pen_color(color::black);
	set_heading_degrees(hour);
	set_pen_width(10);
	draw_distance(150);
}

void ticking_clock()
{	while(true)
	{	set_pen_color(color::white);
		fill_rectangle(0,750,800,800);
		fill_rectangle(0,0,800,100);
		date();
		time();
		double const min = minute();
		double const hr = hour();
		double const sec = second();
		hands(sec, min, hr);
		wait(0.25);  }	}

void main()
{	make_window(600,800);
	draw_circle();
	ticking_clock();
}
```