24780 Engineering Computation: Problem Set 3

(*) In the following instructions (and in all course materials), substitute your Andrew ID wherever you see *yourAndrewId*.

You need to create a ZIP file (which may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas. The filename of the ZIP file must be:

```
PS03-YourAndrewID.zip
```

For example, if your Andrew account is hummingbird@andrew.cmu.edu, the filename must be:

```
PS03-hummingbird.zip
```

Failure to comply with this naming rule will result in an automatic 5% deduction from this assignment's credit. If we cannot identify the submitter of the file, an additional 5% credit will be lost. If we are ultimately unable to connect you with the submitted ZIP file, you will receive 0 points for this assignment. Therefore, ensure strict adherence to this naming rule before submitting a file.

The ZIP file must be submitted to the 24-780 Canvas. If you find a mistake in a previous submission, you can re-submit the ZIP file with no penalty as long as it's before the submission deadline.

Your Zip file should contain only two files, ps3-1.cpp and ps3-2.cpp. Do not include project files and intermediate files generated by the compiler. But, do not worry about some files or directories that are automatically added by the archiver (__MACOSX__ file for example).

Notice: The grade will be assigned to the final submission only. In the case of multiple file submissions, earlier versions will be discarded. Therefore, when resubmitting a ZIP file, it MUST include all the required files. Also, if your final version is submitted after the submission deadline, the late-submission policy will be applied, regardless of how early your earlier version was submitted.

Ensure that your program can be compiled without errors on one of the compiler servers. Do not wait until the last minute, as the compiler servers may become very busy just minutes before the submission deadline!

Submission Due: Please refer to Canvas.

START EARLY!

Unless you are a good programmer, there is no way to finish the assignment overnight.

PS3-1 Cessna 172 Climb Performance [ps3-1.cpp] (30 pts)

Fig. 1 shows a table from an airplane operator's handbook for Cessna 172R. The table tells the maximum climb performance when the airplane is fully loaded.

When we need to know the performance at 0ft altitude (Sea Level), and when the air temperature is 20 Celsius, we immediately know the airplane can climb at 705 ft per minute.

What if the altitude is 1500ft, and temperature is 15 Celsius? We need to interpolate. In this example, first get rate of climb at Sea Level and 15 C, which yields 721 ft/min, and at 2000ft and 15 C, which yields 610 ft/min. Then, interpolate between the two values for 1500 ft, which yields 638 ft/min.

Or, you can interpolate for altitude, then for temperature. Either way will give you the same value.

THe result from this bi-linear interpolation does not give you the exact rate of climb performance, but it is close enough.

Your goal is to write a program that:

1. takes input from the console, altitude and temperature, and
2. if the altitude is less than zero or greater than 10000 (don't worry about above 10000ft), or the temperature is less than -20 or greater than 40, print a message and let the user re-enter the values.
3. then interpolate the climb performance and print the expected maximum rate of climb for the given condition.

By the way, to take modulus of floating point, use fmod function defined in math.h.

Save your source file as ps3-1.cpp and include in the zip file you submit to Canvas.

If you want to save time for typing up the table, you can start from the base code available from Canvas.

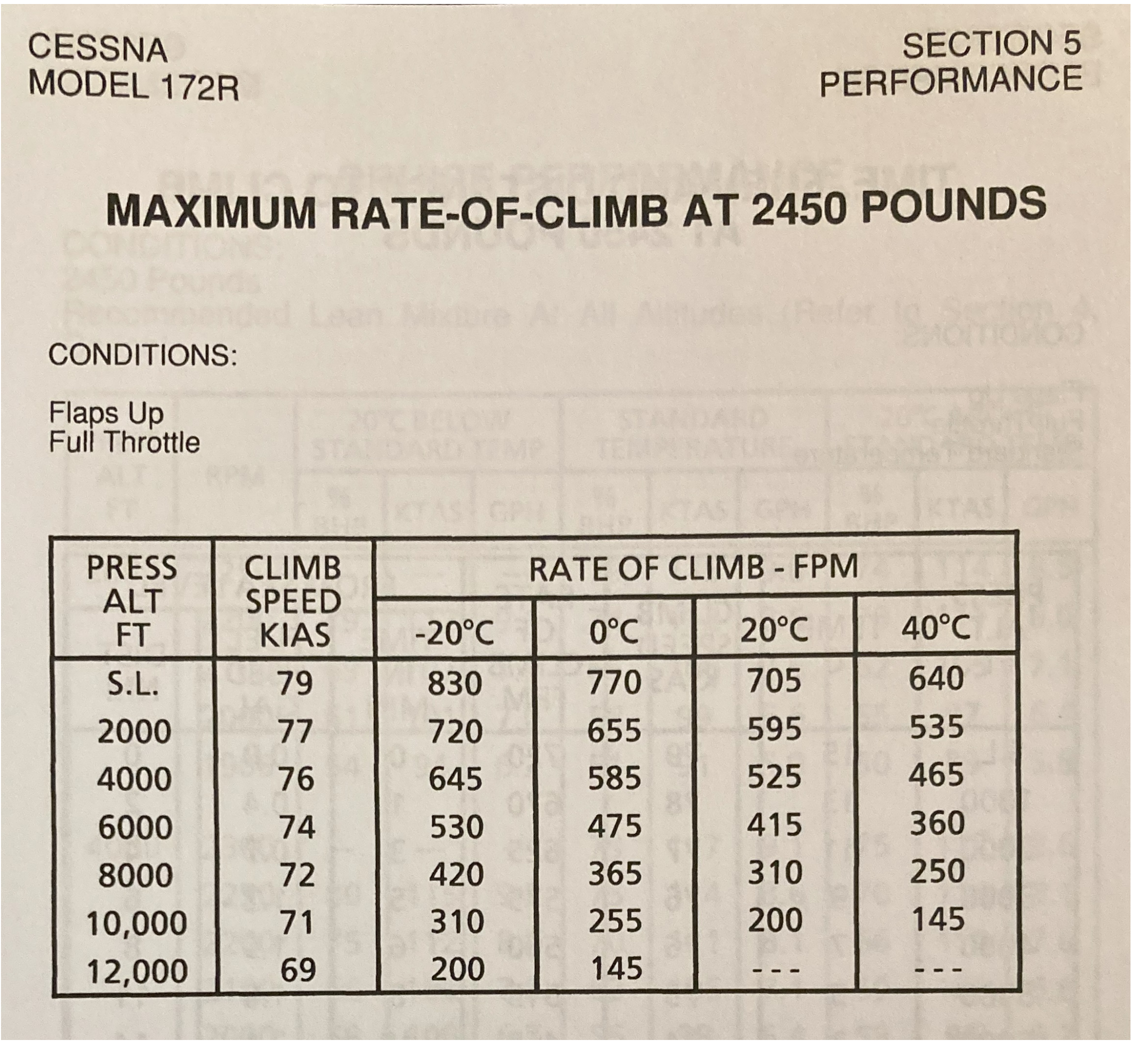
Below is a sample input and output.

```
PS C:\Users\soji\development\teaching\24780\23Fall\ps03> .\ps3-1.exe
Enter Altitude and Temperature:1500 15
Expected Climb Rate=637.812ft/min
```

PS3-2 Wireframe Art [ps3-2.cpp] (70 pts)

When we did not have an image-scanner, pen-tablet, and digital cameras, we drew a picture on a graph paper, or put a transparent graph sheet over a picture, and took coordinates to draw a wire-frame picture on the computer monitor.

Pick your favorite picture, and make it a wire-frame art with OpenGL. You can use any type of OpenGL primitives. Also it does not have to be black & white (but can be black and white.)



CESSNA
MODEL 172R

SECTION 5
PERFORMANCE

MAXIMUM RATE-OF-CLIMB AT 2450 POUNDS

CONDITIONS:

Flaps Up
Full Throttle

PRESS ALT FT	CLIMB SPEED KIAS	RATE OF CLIMB - FPM			
		-20°C	0°C	20°C	40°C
S.L.	79	830	770	705	640
2000	77	720	655	595	535
4000	76	645	585	525	465
6000	74	530	475	415	360
8000	72	420	365	310	250
10,000	71	310	255	200	145
12,000	69	200	145	- - -	- - -

Fig. 1: Cessna 172R Climb Performance Table

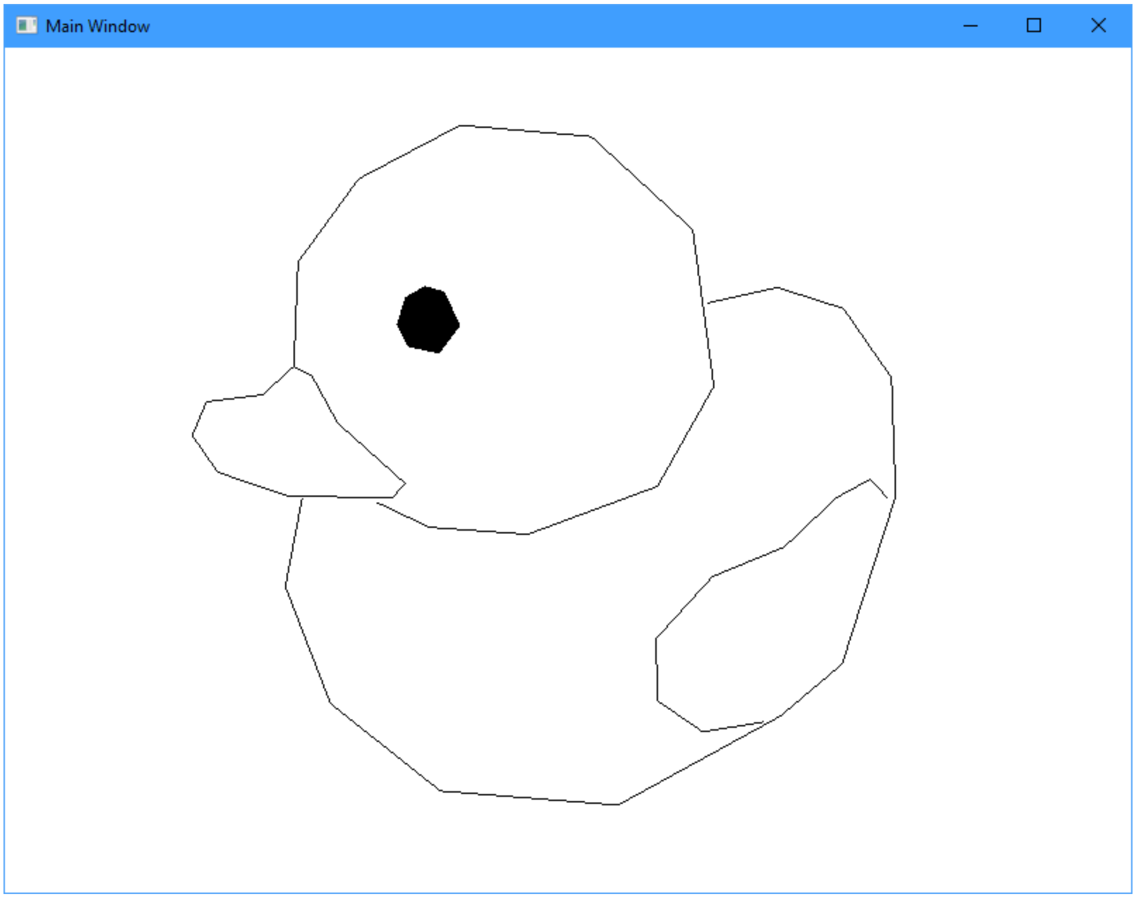


Fig. 2: Sample Wireframe Art - Rubber Duck

Save your program as ps3-2.cpp and include it in the zip file you submit.

Figs. 2 and 3 are the sample pictures.

Test Your Program with One of the Compiler Servers

Test your program with one of the following compiler servers:

```
http://freefood1.lan.local.cmu.edu
http://freefood2.lan.local.cmu.edu
http://freefood3.lan.local.cmu.edu
http://freefood4.lan.local.cmu.edu
```

You need to make sure you are not getting any errors (red lines) from the compiler server.

It is a good practice to remove warnings as well. However, we will not take points off for warnings as long as your program satisfies requirements of the assignment.

You can only access these servers from CMU network. If you need to access from your home, use CMU VPN. Please visit the CMU computing services web site how to install the VPN.

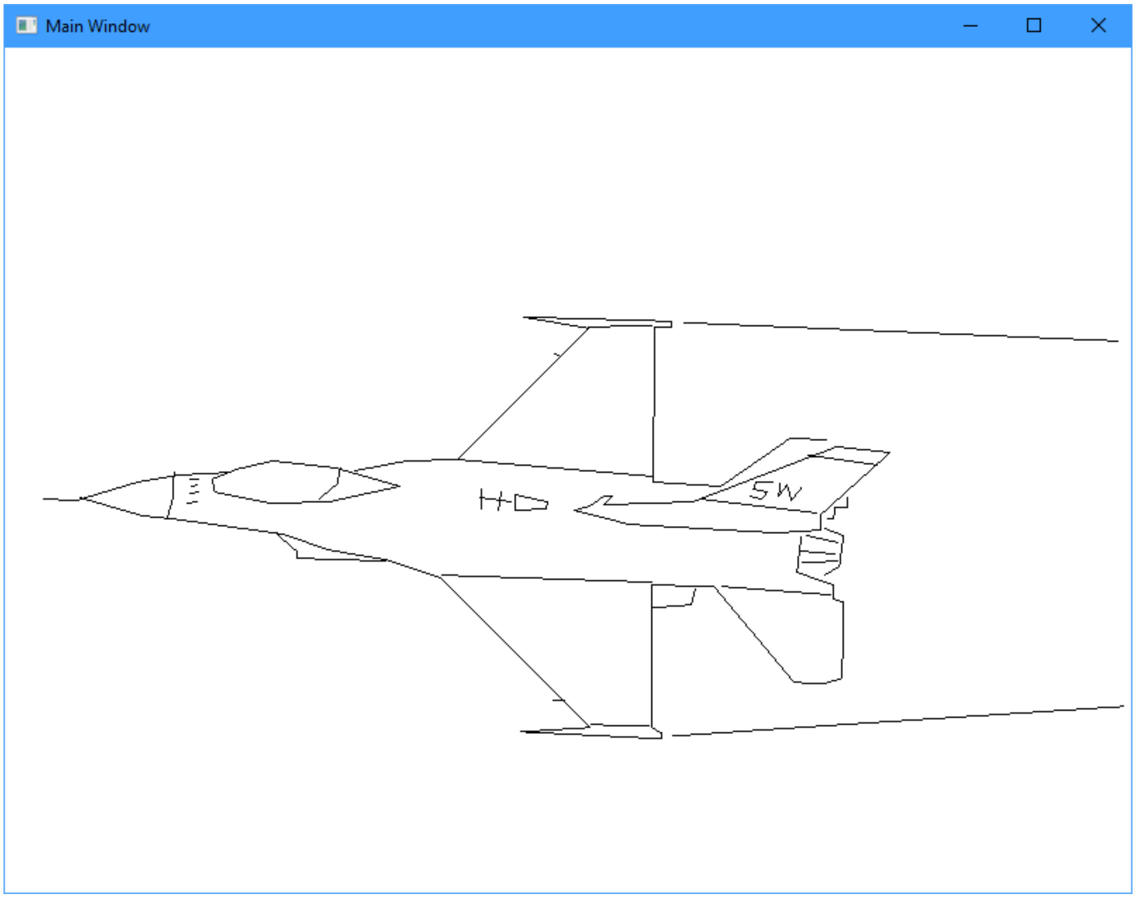


Fig. 3: Sample Wireframe Art - F-16 Fighter Jet