CS 161C++
How to Solve the Input/Output Lab

This is the most complex lab you have had this term. To solve it properly, you need to put together several things: functions, reading from a file, and testing while building.

When looking at a problem like this, I break it into its component parts. In this case, you have three separate programming problems. The first is reading some information from a file. The second is writing a program that makes some choices based on what it reads. The third is creating some geometric shapes on the console.

Creating Geometric Shapes

The first part we are going to consider is how to create the shapes. This is the first thing since it can be done in a series of functions and they can be independently tested.
Since there are several shapes and each is defined by a set of values, it makes sense to write a functions that implement each one. For example, you might have the function

```
void drawRect(char out_char, int rows, int columns)
{
        here you would have code that outputs a shape using out_char
        for example,  drawRect(&,3,2)

                &&
                &&
                &&

}
```

Once you have this function, you could write a main program to test it

```
Int main()
{
        int row, int col;
        char value;
        cout << “Enter your character, row and column values” << endl;
        cin >> value >> row >> col;

        drawRect(value, row, col);

        return 0;

}
```

Now you have a function and you can test it. You can do the same thing for the other shapes:

draw_triangle
draw_square
draw_diamond

Note that each will have multiple loops, since you need to draw a number of rows and each row will have multiple copies of the output character on them.

Making program choices based on input

The next piece to consider is how to design the main program so that it properly chooses the correct function and calls it with the right values.

When designing this, consider the previous lab where you had a program that asked the user for a code and then, based on the choice of code, asked the user for some values and computed the areas of the shapes.

A similar approach would work here.

Now that you have your drawing functions defined above, you could write a program that would ask the user for an input code “T”, “D”, “S”, “R”, “E” and based on the input code you would then ask for a drawing character and dimensions. Now call the appropriate function. Repeat this until the user enters “E”.

Now you have a program that has the proper logic and tests the drawing functions and you are ready for the final part.

Reading from a file

The first step in reading from a file is to write a short test program that opens the file, reads each line and displays it, and then closes the file. This shows that you are able to open and read from the file.

Once you have done this, you can then combine all three parts in one program:

1) Open the input file
2) Read the information from the file instead of asking the user for it
3) Call the drawing methods