CMSC 335 - Project 2

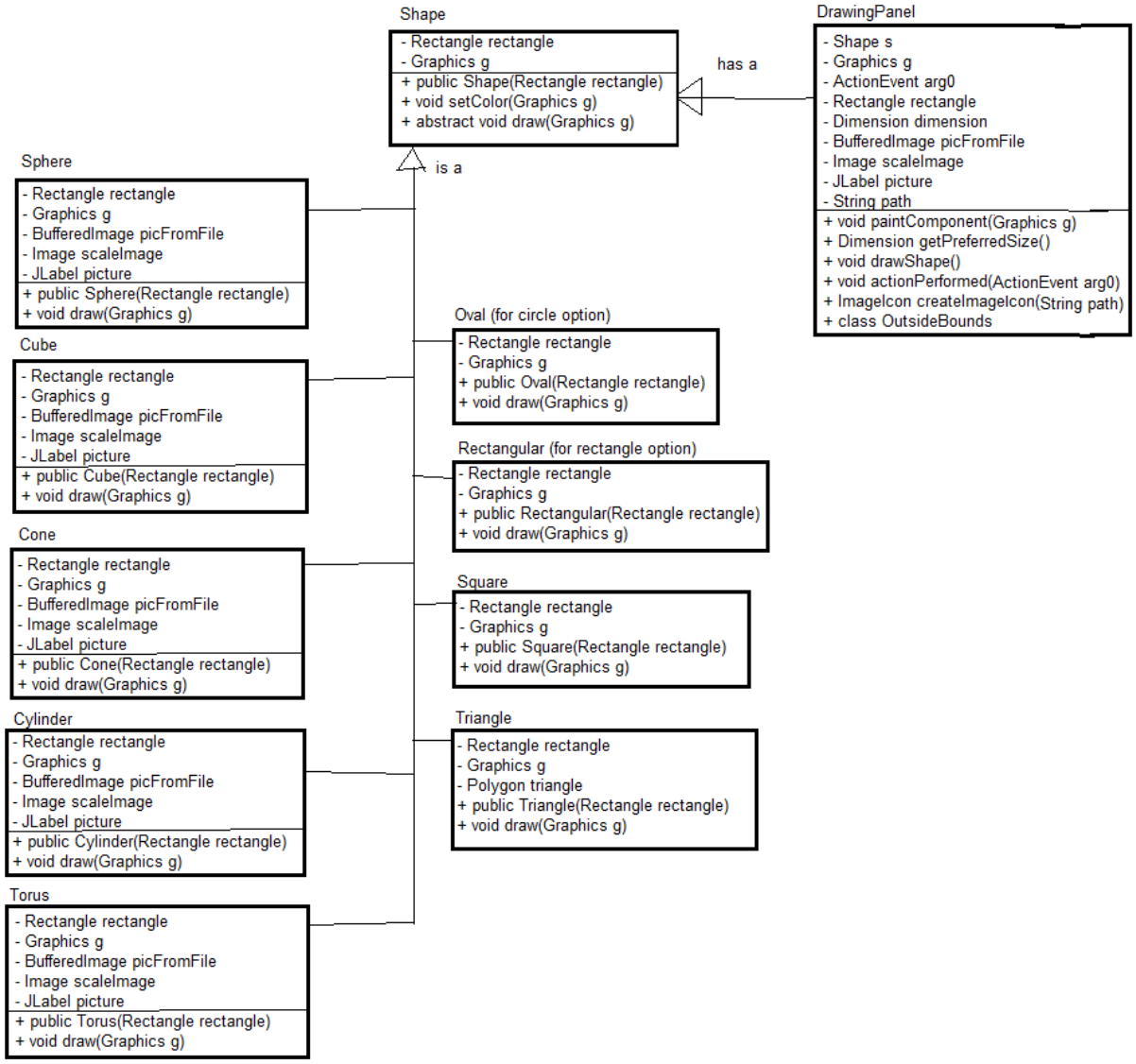
Java Shape Drawings

Angel Lee

Feb 15, 2021

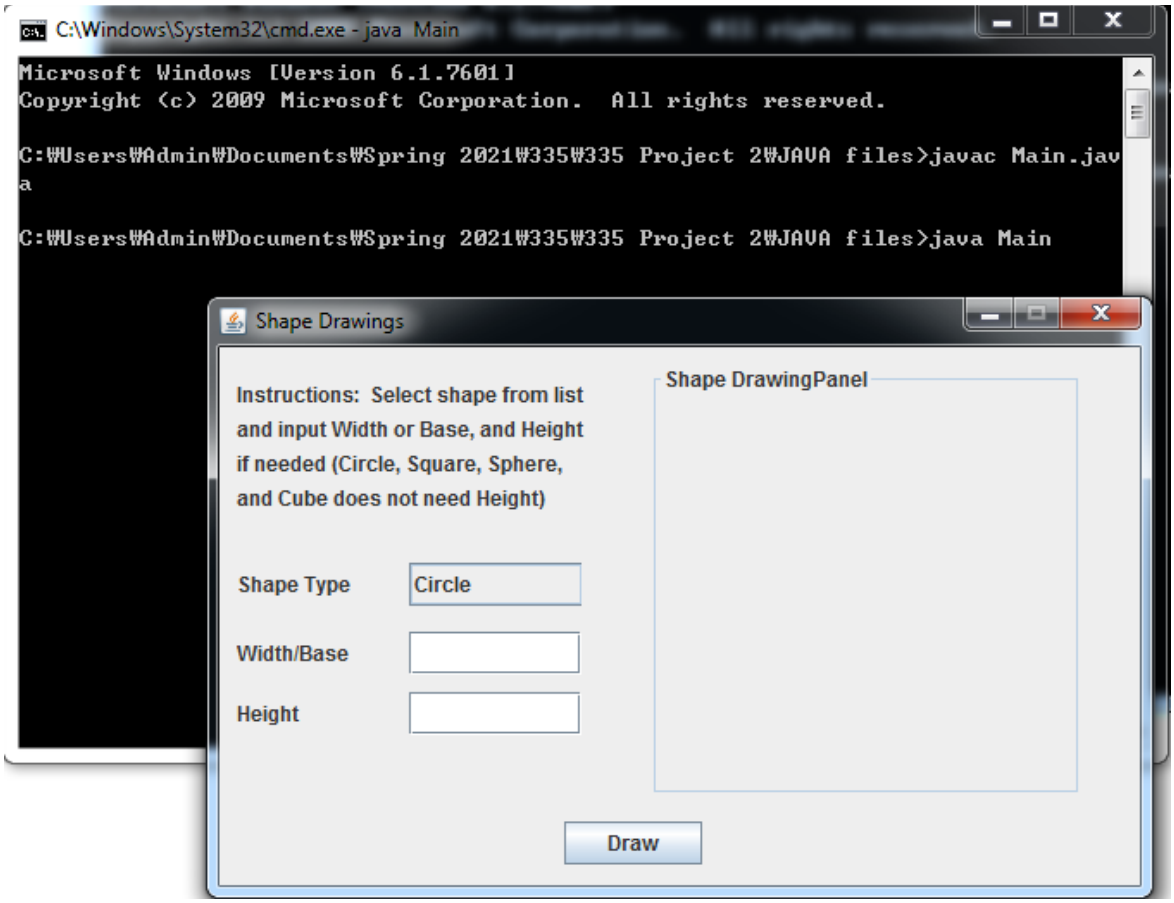
UML Diagram:

UML Class diagram of my program demonstration "is a" relationship (separate PNG file also in zip) Shape is the main Object here, while the other classes are "oriented" towards it, which is why it's called "object orientated programming", and the DrawingPanel has a (composed) of Shape:

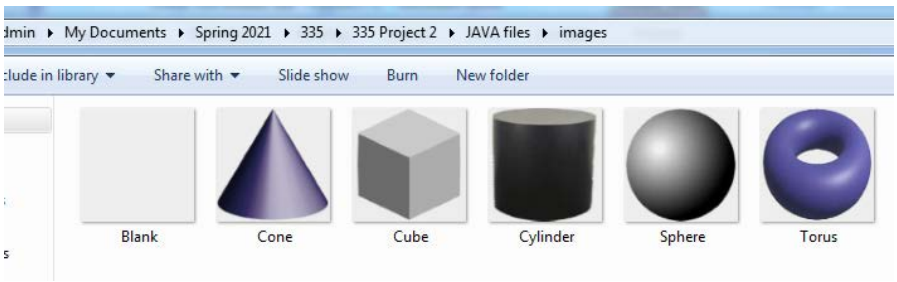


Compilation on Command Prompt/Java SDK:

When opening this with the command prompt compiler, we first make the java classes first and then pull up the Main class. I type "cmd" onto the address bar to pop the command prompt right in the folder that I want. The result:



Here are the 3D images for 3D shapes in the folder:



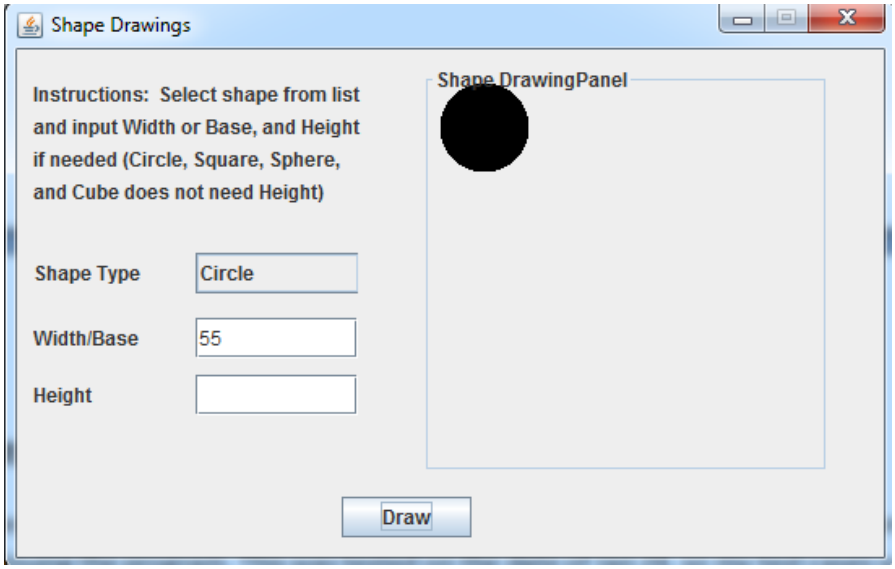
Test cases:

I have made 10 cases testing different shape inputs and yes/no/other when it comes to continuing the program. This was tested on the date of Jan 24, so my test cases will show that date and time (around 7pm) but should provide the local time of other testers who try this.

Test case:	Input:	Expected Output:	Pass?
1	Circle at Width of 55, blank Height (putting anything just clears it)	Circle drawing	Y
2	Circle with base "b" and height 55	ERROR saying input cannot be blank or non-number	y
3	Square at Width 55, Height 55	Square drawing, and cleared Height (since unnecessary)	Y
4	Square with Width 300	ERROR saying it exceeds Panel border	Y
5	Triangle at Width/Base 55, blank Height	ERROR saying height cannot be blank or non-number	Y
6	Triangle at Width/Base 55, height 22	Triangle	Y
7	Rectangle Width 55, Height 22	rectangle	Y
8	Rectangle blank Height input	ERROR saying Height cannot be blank	Y
9	Sphere with width of 100	Sphere	Y
10	Sphere with width 250, which exceeds Panel border	ERROR saying it exceeds Panel border	y
11	Cube with width/base 200	Cube	y
12	Cube, blank input	ERROR saying input must be integers	Y
13	Cone with width/base 200 and no height	ERROR saying height cannot be blank or non-number	y
14	Cone with base 200 and height 200	Cone	y
15	Cylinder with base 55 and height 200	Very skinny cylinder	y
16	Cylinder with height "b"	ERROR saying height cannot be blank or non-number	y
17	Torus with base 200 and height "b"	ERROR saying height cannot be blank or non-number	y
18	Torus with base 100 and height 50	Torus	y

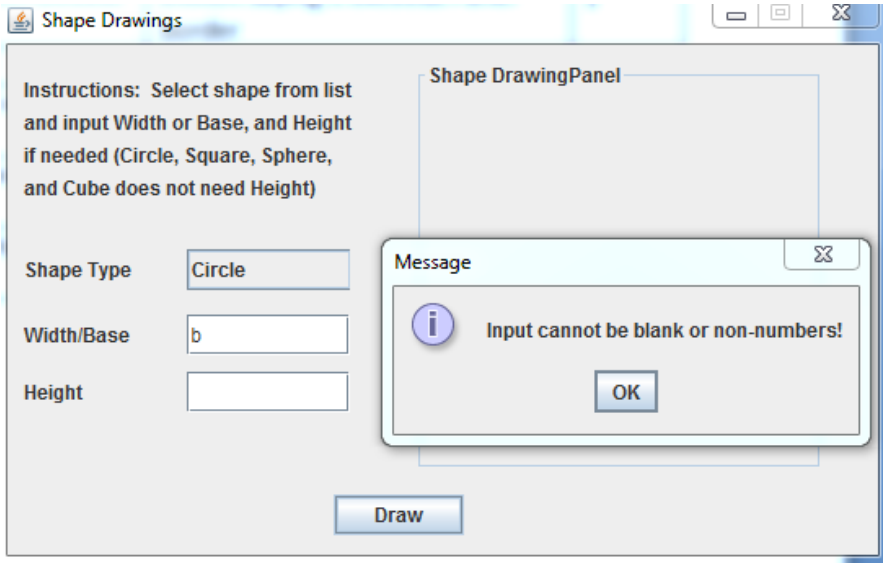
Test case 1:

Checking option "circle" and using the width/base text field, there is a result with no error:



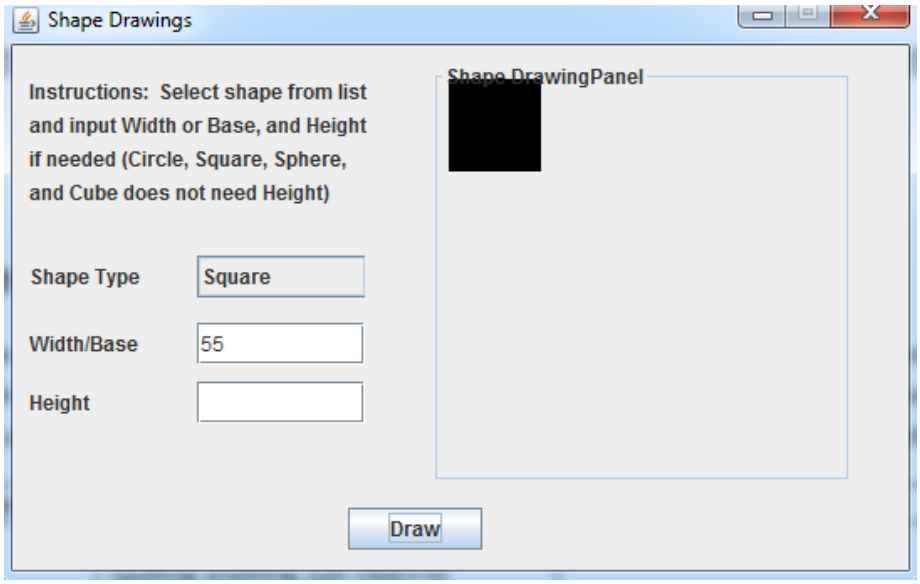
Test case 2:

Back to circle, which only requires a width input. With a non-number input, there's an error.



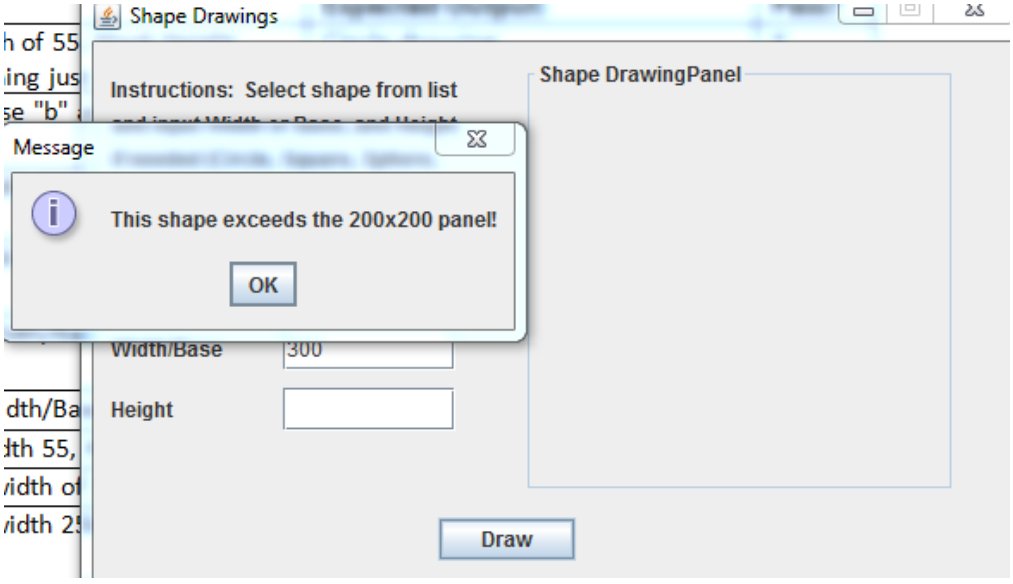
Test case 3:

Same with the square, which also only requires Width input. Height was inputted but clears as it's unnecessary.



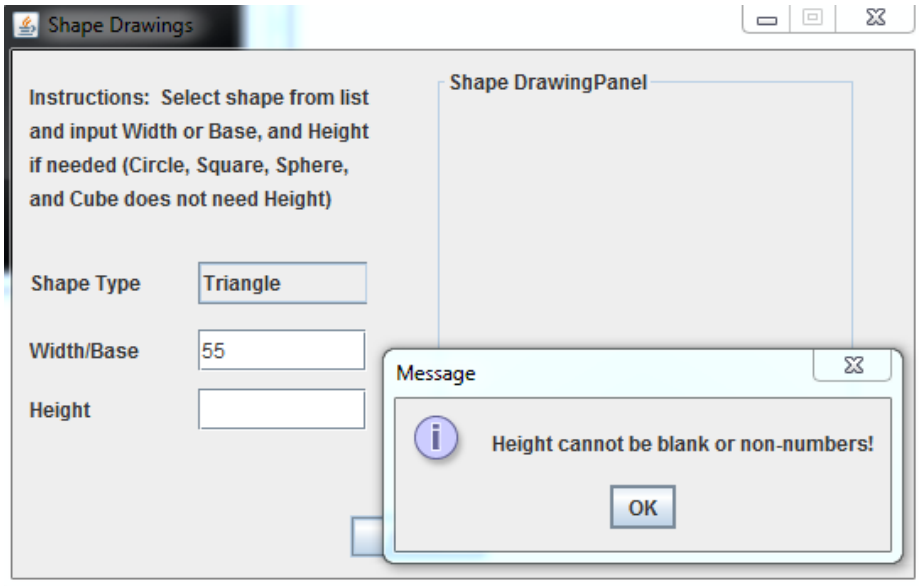
Test case 4:

Input of 300 for width/base and the Square gives an error.



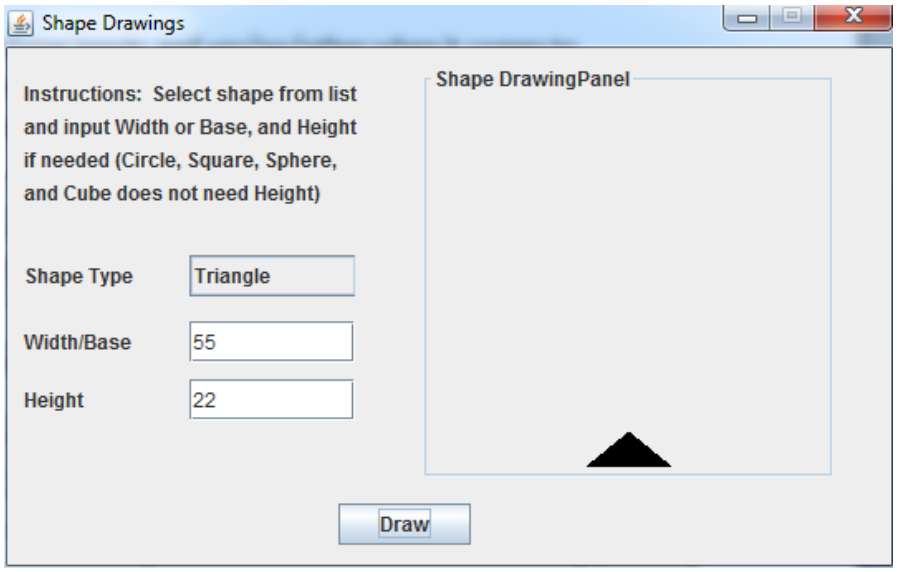
Test case 5:

For triangle unlike the previous two options, requires a Height. When left blank, it declares an error:



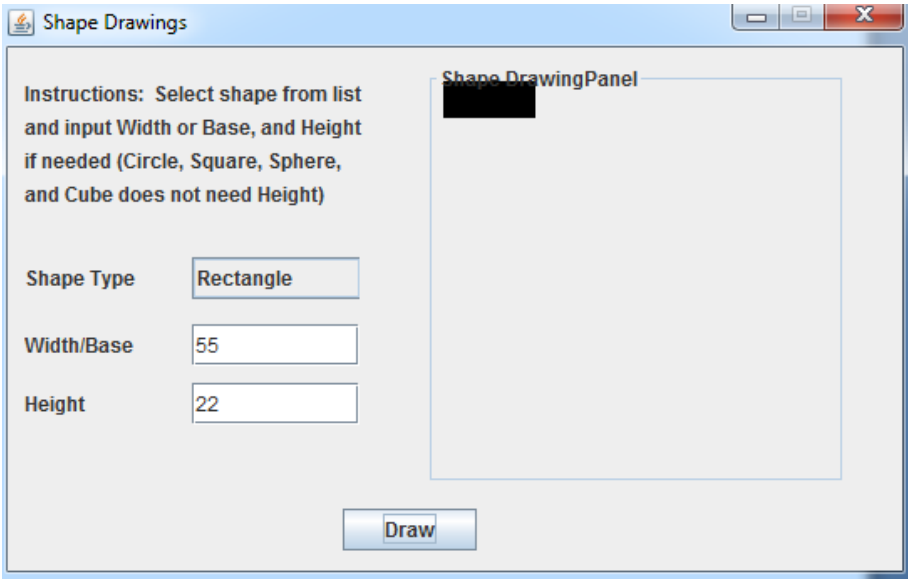
Test case 6:

With the proper Height input, it compiles correctly.



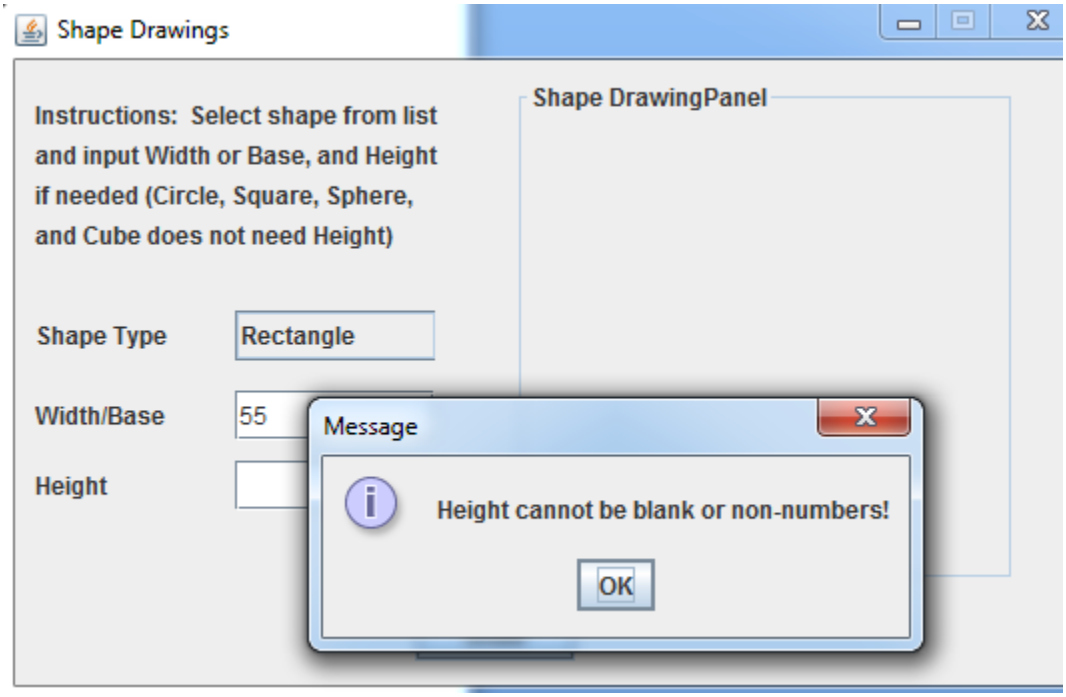
Test case 7:

Rectangle compiles correctly with the proper input.



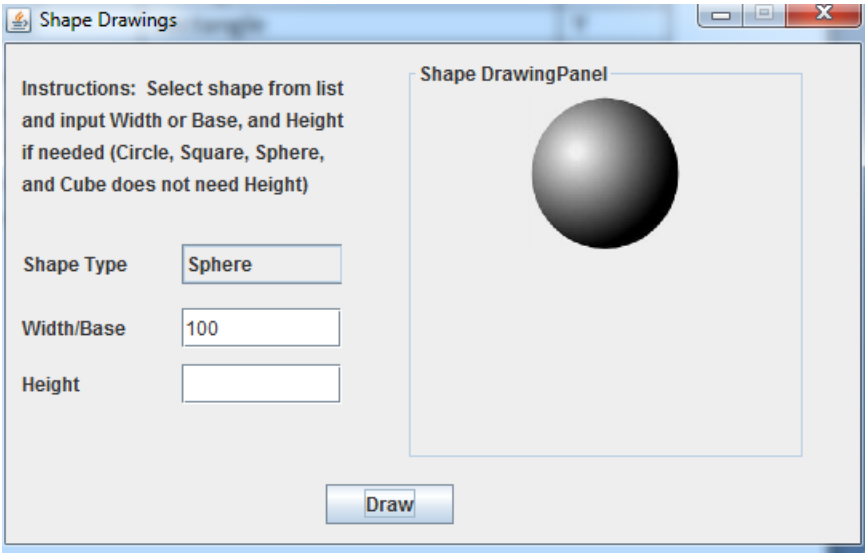
Test case 8:

No height input for Rectangle will give an ERROR



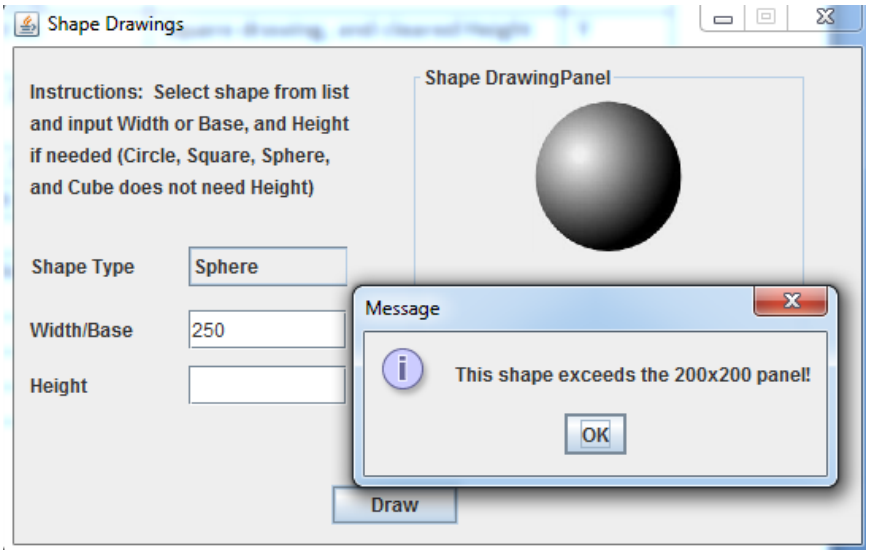
Test case 9:

Sphere compiles correctly with proper input



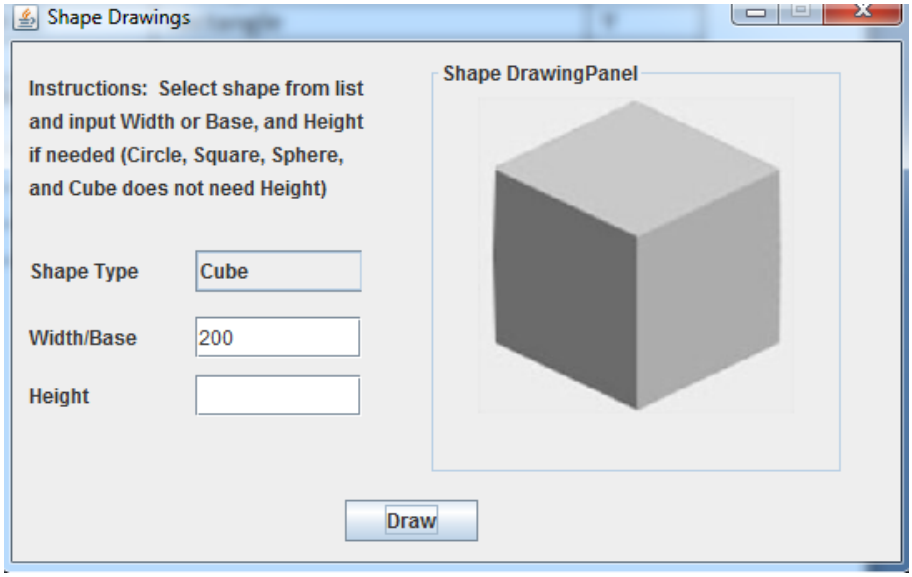
Test case 10:

Dimension input was large, and it gives an error stating it exceeds the panel.



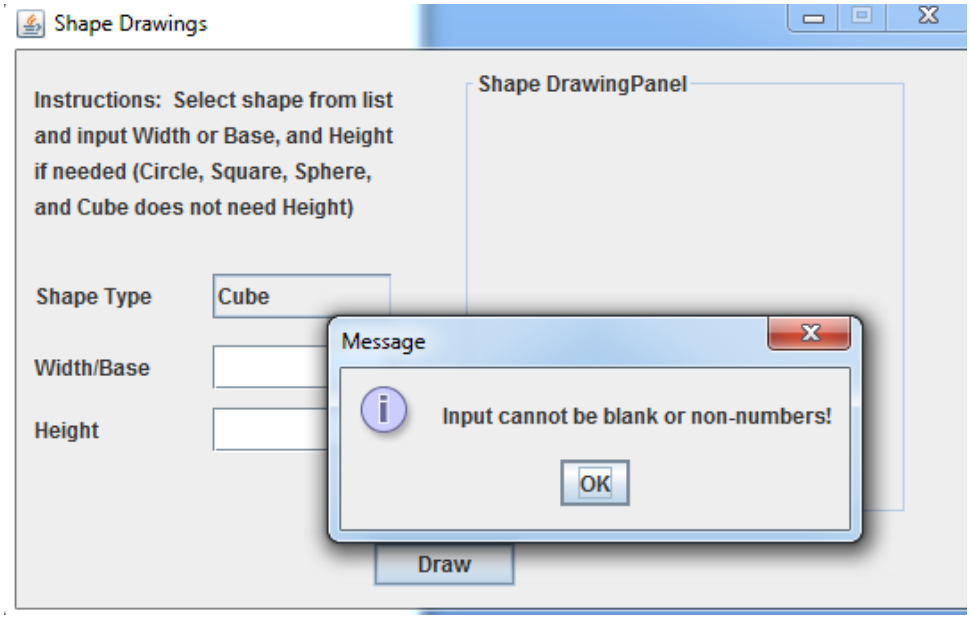
Test case 11:

Cube compiles correctly with the properly sized Base input.



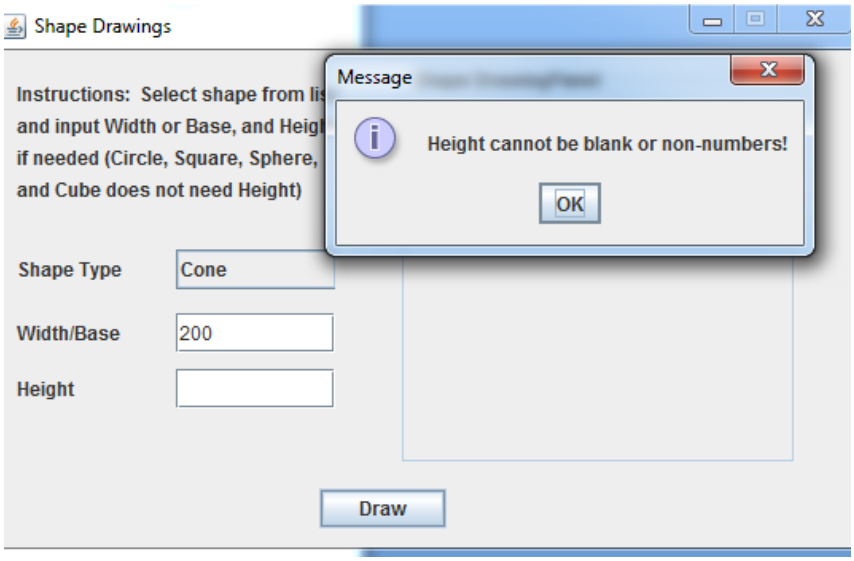
Test case 12:

Blank input gives an error for Cube



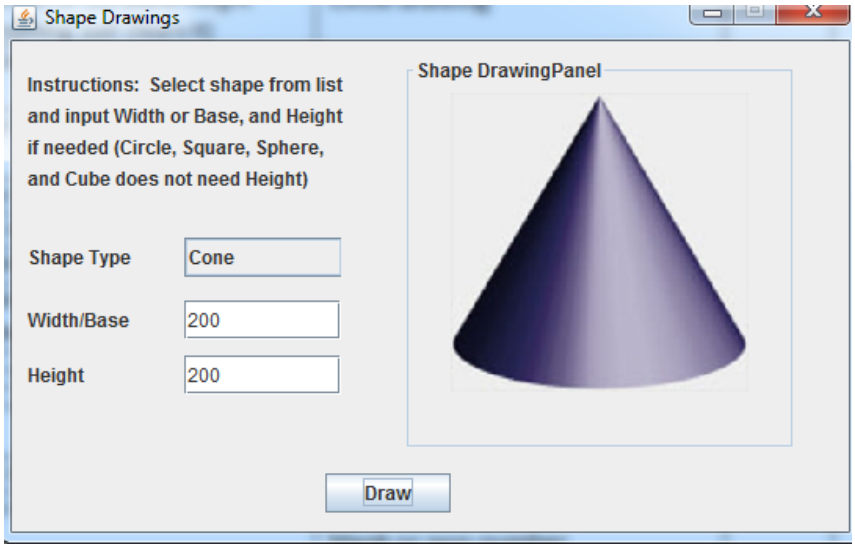
Test case 13:

Cone requires a height input, and gives an error when left blank.



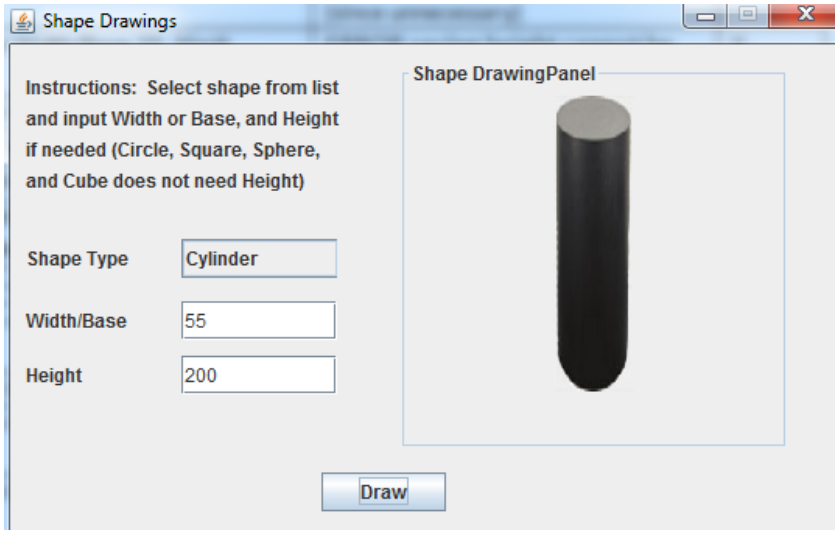
Test case 14:

Cone compiles properly with both base and height input



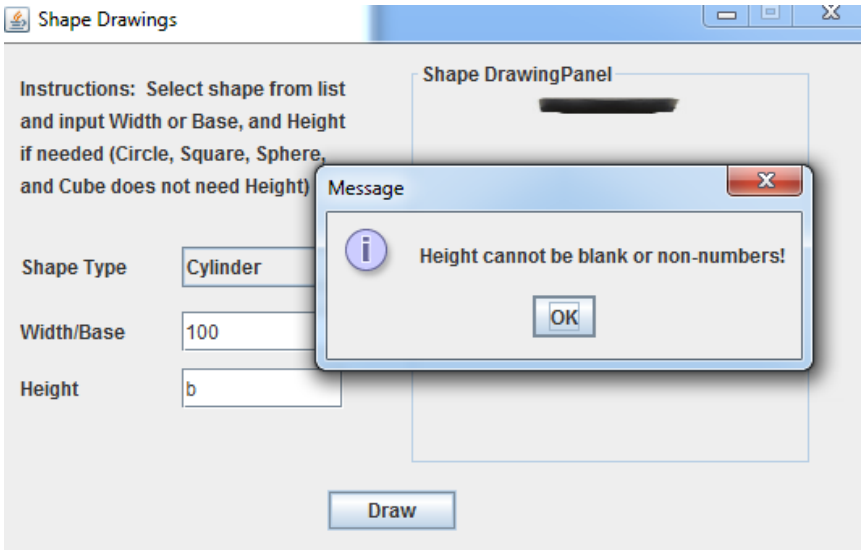
Test case 15:

Give the Cylinder a small base and tall height, it makes a skinny cylinder.



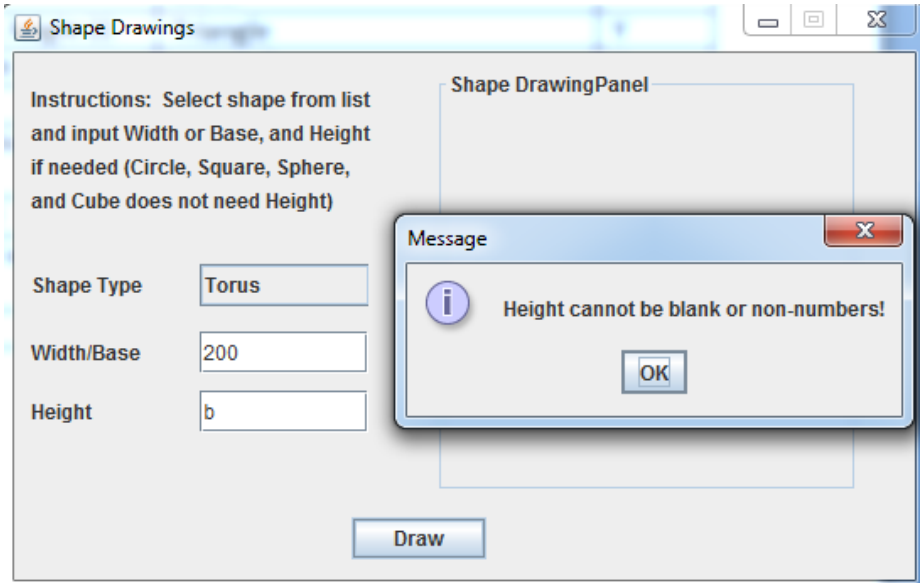
Test case 16:

Give the height a bad input, says height cannot be non number



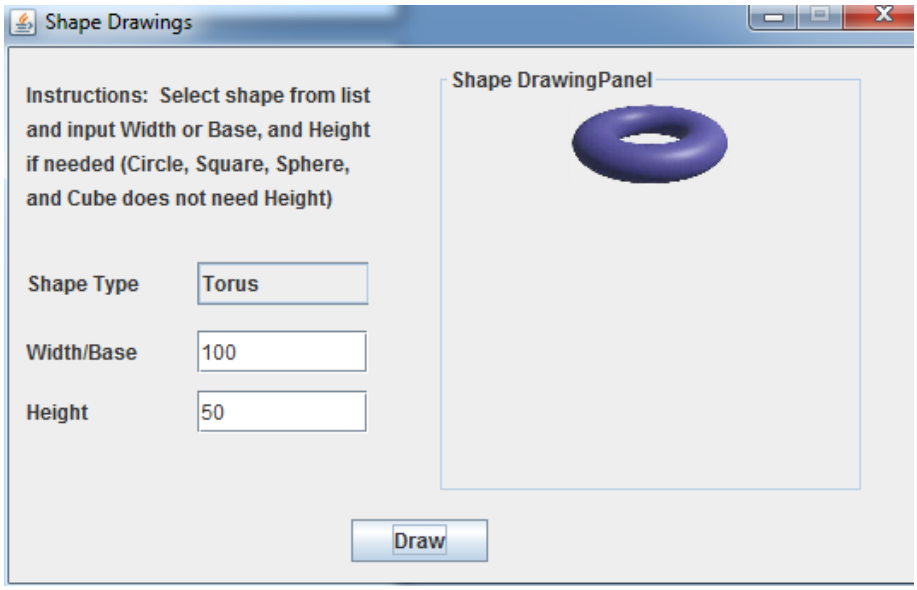
Test case 17:

Torus requires a Height, and inputting a non-number shoots an error.



Test case 18:

Provide height and width for Torus and it compiles



Conclusion:

The program has successfully done all the test cases as expected. The program didn't take me as long as expected, and I believe it was because I was allowed to use an image file to represent 3D shapes.

What I learned from this program is how to use the Paint classes to create shapes and also call image files onto the GUI. It was a bit difficult as I had to figure out how to make the image represent only in the panel and NOT in the general Frame, and how to clear the images and previous shapes out. I noticed that while making the program, if I had a Shape object, it wouldn't clear out when I called an Image object, so as I called an Image object, I also had to call a blank Shape object (in the class of blankShape) at the same time to replace the previous Shape object.

I liked that I had more freedom regarding how to design this program, especially with the presentation - the GUI and frames and the inputting options - compared to previous projects I've done in school.

Regarding your comment: I think I understand the "idea" of object oriented programming, but trying to put it together into a functioning and elegant code is another. I even tried separating the Shape abstract classes into 2D shape and 3D shape classes to prevent copy/pasting duplicate code for the 3D shapes but I wasn't able to figure out how to make it work, it seems to compile weird with the Draw button for some reason. I likely would've figured it out with more time, but hopefully though this revision fits the criteria.