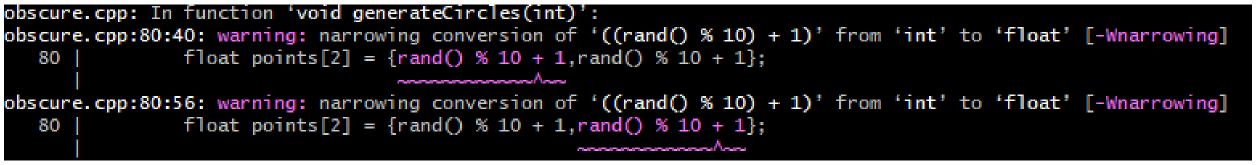
Unfortunately, I was unable to completely get the circles to work properly. While the points and data were created from the code correctly, the outputted information is incorrect with the Angle in which the rectangle obscures not existing. No graphical representation is present as the Rectangles were not generating code-wise properly. Two warnings appear, stating that a conversion of INT to FLOAT is occurring, but does not alter the outcome of the data.



```
obscure.cpp: In function 'void generateCircles(int)':
obscure.cpp:80:40: warning: narrowing conversion of '((rand() % 10) + 1)' from 'int' to 'float' [-Wnarrowing]
   80 |         float points[2] = {rand() % 10 + 1,rand() % 10 + 1};
      |                            ~~~~~~~~~~~~^~~
obscure.cpp:80:56: warning: narrowing conversion of '((rand() % 10) + 1)' from 'int' to 'float' [-Wnarrowing]
   80 |         float points[2] = {rand() % 10 + 1,rand() % 10 + 1};
      |                                            ~~~~~~~~~~~~^~~
```

The Rectangle information is then outputted to the 'rectData.txt' file. The listed information within the file are as follows:

$Center_X$, $Center_Y$, P_{1x}, P_{1y}, P_{2x}, P_{2y}, P_{3x}, P_{3y}, P_{4x}, P_{4y}, $Angle_{min}$, $Angle_{Max}$

Since I was unable to compute the angles of obscurity, my time complexity result is unsolved. The assumption of my time complexity for determining the angle obscure would be Big-O(N), as I would run through a sorted array of rectangles and determine through two loops the obscurity of the angle and any rectangle behind it.