Team Members
160050068 -> Karan Kumar
160050069 -> Shantanu Kumar

Citations: code for bresenham's algorithm was written by taking help from here:
https://gist.github.com/nowke/965fed0d5191bf373f1262be584207bb

Keyboard Controls:
+ -> increases the brush as well as eraser size by 2 with one click over all the modes
- -> decreases the brush as well as eraser size by 2 with one click over all the modes
// Each point is a circle of the current size as the radius

P -> switch between modes
B -> switch to brush
E -> switch to eraser
F -> switch to fill
C -> change brush color // takes four inputs from console r g b a
D -> change background color i.e. eraser color

For increasing point size:
'+' Increases the size of the brush and then stroke makes a circle of the given size as radius and radius of circle 1 means a single point.

For the Smooth Brush:
--We have implemented the Smooth Brush Class(smooth_brush_t inside the brush.cpp & brush.hpp files) with smooth brush and smooth eraser. With their separate stroke functions similar to the point brush class. The stroke function takes two points and uses bresenham's algorithm to fill pixels in the line between the two points.
--Mouse move callback function which is called when the mouse is dragged. If you press the left button of the mouse and drag it then the smooth brush feature works and a line is drawn between the two points given by the mouse move callback function is drawn.

Primitive Modes:
--The other primitive modes are implemented in primitive.cpp and .hpp
--We have state variables and a draw function which calls the smooth brush strokes function to draw lines according to the state.
--The keyboard controls are added to the keyboard callback.

Fill:
Implemented in fill.cpp.
Used a queue to perform BFS to do flood fill.
Several small changes in brush.cpp canvas.cpp primitive.cpp fill.cpp gl_framework.cpp and brush.hpp canvas.hpp primitive.hpp fill.hpp gl_framework.hpp to implement above features