

**KHAIRULLAH KHALIQ**  
**04072113003**  
**COMPUTER GRAPHICS**  
**ASSIGNMENT#3**  
***SUBMITTED***  
***TO***  
***MAM MEMOONA AFSHEEN***

## Introduction:

This assignment involved completing a skeleton OpenGL project by implementing:

- Line drawing using the Midpoint Line Algorithm
- Circle drawing using the Midpoint Circle Algorithm
- Flood fill algorithm
- Script to create artwork
- Event handling for mouse and keyboard

### 1. Draw Line Function:

- I implemented the Midpoint Line Drawing Algorithm.
- It calculates error terms to decide whether to move horizontally, vertically, or diagonally at each step.
- It only uses **integer arithmetic** (no floating-point operations).
- The line is drawn by repeatedly plotting points from the start (x0, y0) to the end (x1, y1).

### 2. Draw Circle Function:

- I implemented the Midpoint Circle Drawing Algorithm.
- It draws one-eighth of the circle and uses symmetry to plot the other seven parts.
- Again, no floating-point arithmetic is used.

### Key Steps:

- Start at (0, R)
- Move outward while updating the decision variable  $d$
- Plot all symmetrical points in the eight octants

### 3. Flood Fill Function:

- I implemented flood fill using a **queue** instead of recursive calls to avoid stack overflow.
- It colors all connected pixels that have the same color as the starting pixel.

### Key Points:

- First, the function finds the color at the clicked pixel.
- Then it changes all matching neighbor pixels to the new color.

### 4. Artwork:

- I created a **flower drawing** using multiple circles and lines.
- The flower has:
  - A golden center
  - Pink petals around it
  - A green stem
  - Small leaves using tiny circles



*"Spring Bloom"*

# Challenges I Faced

## 1. OpenGL Setup:

- It was tricky to configure GLEW and FreeGLUT libraries correctly. It takes 5 hours to setup.
- Sometimes the window would not open and gives error because of missing .dll files.

## 2. Flood Fill Issues:

- Initially, I tried recursive flood fill and hit a **stack overflow error** for larger areas.
- I fixed it by switching to an **iterative flood fill** using a queue.

## 3. Understanding Mouse Coordinates:

- I had to adjust mouse coordinates because OpenGL and window y-coordinates are flipped.

## 4. Fat Circle Not Implemented Fully:

- I ran out of time to create a separate fat circle function (optional task), but I made the artwork using multiple circles creatively instead.

## 5. Performance on Large Images:

- Flood filling big areas sometimes made the window freeze for a few seconds .(most challenging )