# CSC 322 Introduction to Computer Graphics Spring 2025

## Homework #3

**Due Date:** 02/12/2025 before class
**Submission Link:** https://learn.cua.edu

Course textbook (pages 50-51): Problems 3, 4, 5, 6, 8, 10

**Question A1:** Modify the provided Threejs code from Lecture 7 (which draws a circle) in JSFiddle to visualize the implicit equation.

- $F(x, y) = (x^2 + y^2)^2 - 2^2(x^2 - y^2) = 0$

Once you've finished, upload your code to GitHub and include a screenshot of your drawing with your homework assignment.

```
// Import THREE
import * as THREE from
'https://threejsfundamentals.org/threejs/resources/threejs/r132/build/three.module.js';

// Import OrbitControls
import { OrbitControls } from
'https://cdn.jsdelivr.net/npm/three@0.121.1/examples/jsm/controls/OrbitControls.js';

// Create renderer
const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

// Create camera
const camera = new THREE.PerspectiveCamera(45, window.innerWidth /
window.innerHeight, 1, 500);
camera.position.set(0, 0, 100);
```

```
// Create OrbitControls
const controls = new OrbitControls(camera, renderer.domElement);
controls.enableDamping = true; // Adds smooth motion
controls.dampingFactor = 0.05;
controls.screenSpacePanning = false;
controls.minDistance = 10;
controls.maxDistance = 200;

// Create scene
const scene = new THREE.Scene();

// Create a circle
const circleMaterial = new THREE.LineBasicMaterial({ color: 0xff0000 });
const circlePoints = [];
const radius = 25;  // Circle radius
const segments = 100;  // Number of segments to approximate the circle

for (let i = 0; i <= segments; i++) {
  const theta = (i / segments) * 2 * Math.PI;
  const x = radius * Math.cos(theta);
  const y = radius * Math.sin(theta);
  circlePoints.push(new THREE.Vector3(x, y, 0));
}

const circleGeometry = new THREE.BufferGeometry().setFromPoints(circlePoints);
const circle = new THREE.Line(circleGeometry, circleMaterial);
scene.add(circle);

// Animation loop for rendering and controls update
function animate() {
  requestAnimationFrame(animate);
  controls.update();  // Required for damping to work
  renderer.render(scene, camera);
}

// Start the animation loop
animate();
```