MACHAKOS UNIVERSITY

University Examinations for 2018/2019

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTING AND INFORMATION TECHNOLOGY

THIRD YEAR SECOND SEMESTER EXAMINATION FOR

BACHELOR OF SCIENCE (MATHEMATICS AND COMPUTER SCIENCE)

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

SCO 305: COMPUTER GRAPHICS

DATE: 17/4/2019 **TIME: 11.00-1.00 PM**

INSTRUCTIONS

Answer **question ONE and any other TWO questions.**

QUESTION ONE (30 MARKS) (COMPULSORY)

a) Explain four aspects of an object that changes due to transformation (4 marks)

b) Differentiate between: (10 marks)

i. 2D and 3D

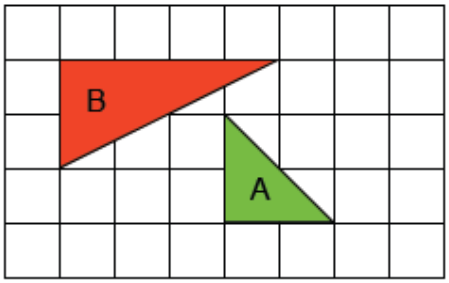
ii. Frame buffer and resolution.

iii. Conformal and Affine

iv. Computer Graphic Metafile and Programmer's Hierarchical Interactive Graphics System (PHIGS)

v. B-Spline curves and Spline curves

c) Describe the transform needed to transform the triangle from A to B in Figure below (6 marks)



d) Draw the output of the OpenGl code shown below: (10 marks)

```
#include <windows.h>
#include <GL/glut.h>

char title[] = "3D Shapes";

void initGL() {
  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
  glClearDepth(1.0f);
  glEnable(GL_DEPTH_TEST);
  glDepthFunc(GL_LEQUAL);
  glShadeModel(GL_SMOOTH);
  glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
}

void display() {
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glMatrixMode(GL_MODELVIEW);

  glLoadIdentity();
  glTranslatef(1.5f, 0.0f, -7.0f);
  glBegin(GL_QUADS);
    glColor3f(0.0f, 1.0f, 0.0f);
    glVertex3f( 1.0f, 1.0f, -1.0f);
    glVertex3f(-1.0f, 1.0f, -1.0f);
    glVertex3f(-1.0f, 1.0f,  1.0f);
    glVertex3f( 1.0f, 1.0f,  1.0f);


    glColor3f(1.0f, 0.5f, 0.0f);     // Orange
    glVertex3f( 1.0f, -1.0f,  1.0f);
    glVertex3f(-1.0f, -1.0f,  1.0f);
    glVertex3f(-1.0f, -1.0f, -1.0f);
    glVertex3f( 1.0f, -1.0f, -1.0f);


    glColor3f(1.0f, 0.0f, 0.0f);     // Red
    glVertex3f( 1.0f,  1.0f, 1.0f);
    glVertex3f(-1.0f,  1.0f, 1.0f);
    glVertex3f(-1.0f, -1.0f, 1.0f);
    glVertex3f( 1.0f, -1.0f, 1.0f);

    glColor3f(1.0f, 1.0f, 0.0f);     // Yellow
    glVertex3f( 1.0f, -1.0f, -1.0f);
    glVertex3f(-1.0f, -1.0f, -1.0f);
    glVertex3f(-1.0f,  1.0f, -1.0f);
    glVertex3f( 1.0f,  1.0f, -1.0f);

    glColor3f(0.0f, 0.0f, 1.0f);     // Blue
    glVertex3f(-1.0f,  1.0f,  1.0f);
```

```
      glVertex3f(-1.0f,  1.0f, -1.0f);
      glVertex3f(-1.0f, -1.0f, -1.0f);
      glVertex3f(-1.0f, -1.0f,  1.0f);


      glColor3f(1.0f, 0.0f, 1.0f);     // Magenta
      glVertex3f(1.0f,  1.0f, -1.0f);
      glVertex3f(1.0f,  1.0f,  1.0f);
      glVertex3f(1.0f, -1.0f,  1.0f);
      glVertex3f(1.0f, -1.0f, -1.0f);
   glEnd();

   glLoadIdentity();
   glTranslatef(-1.5f, 0.0f, -6.0f);
   glBegin(GL_TRIANGLES);
      glColor3f(1.0f, 0.0f, 0.0f);     // Red
      glVertex3f( 0.0f, 1.0f, 0.0f);
      glColor3f(0.0f, 1.0f, 0.0f);     // Green
      glVertex3f(-1.0f, -1.0f, 1.0f);
      glColor3f(0.0f, 0.0f, 1.0f);     // Blue
      glVertex3f(1.0f, -1.0f, 1.0f);

      glColor3f(1.0f, 0.0f, 0.0f);     // Red
      glVertex3f(0.0f, 1.0f, 0.0f);
      glColor3f(0.0f, 0.0f, 1.0f);     // Blue
      glVertex3f(1.0f, -1.0f, 1.0f);
      glColor3f(0.0f, 1.0f, 0.0f);     // Green
      glVertex3f(1.0f, -1.0f, -1.0f);

      glColor3f(1.0f, 0.0f, 0.0f);     // Red
      glVertex3f(0.0f, 1.0f, 0.0f);
      glColor3f(0.0f, 1.0f, 0.0f);     // Green
      glVertex3f(1.0f, -1.0f, -1.0f);
      glColor3f(0.0f, 0.0f, 1.0f);     // Blue
      glVertex3f(-1.0f, -1.0f, -1.0f);

      glColor3f(1.0f,0.0f,0.0f);       // Red
      glVertex3f( 0.0f, 1.0f, 0.0f);
      glColor3f(0.0f,0.0f,1.0f);       // Blue
      glVertex3f(-1.0f,-1.0f,-1.0f);
      glColor3f(0.0f,1.0f,0.0f);       // Green
      glVertex3f(-1.0f,-1.0f, 1.0f);
   glEnd();

   glutSwapBuffers();
}

void reshape(GLsizei width, GLsizei height) {
   if (height == 0) height = 1;
   GLfloat aspect = (GLfloat)width / (GLfloat)height;
```

```
    glViewport(0, 0, width, height);

  glMatrixMode(GL_PROJECTION);  glLoadIdentity();    gluPerspective(45.0f, aspect, 0.1f,
100.0f);
}

int main(int argc, char** argv) {
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_DOUBLE);   glutInitWindowSize(640, 480);
glutInitWindowPosition(50, 50);
  glutCreateWindow(title);
  glutDisplayFunc(display);
  glutReshapeFunc(reshape);
  initGL();
  glutMainLoop();
  return 0;
}
```

QUESTION TWO (20 MARKS)

a) Draw on Cartesian plane the image drawn by code segment below (Note that the coordinate system has been added for illustration.) (6 marks)

Given is the following code segment:

```
glBegin(GL_TRIANGLE_STRIP);
glColor3f(0.2, 0.2, 0.2);
glVertex3f(0, 0, 0);
glColor3f(0.8, 0.8, 0.8);
glVertex3f(0, 2, 3);
glVertex3f(3, 2, 0);
glColor3f(0.2, 0.2, 0.2);
glVertex3f(0, 4, 0);
glEnd();
```

b) Describe the responsibility of the following state of graphics pipeline (6 marks)

   i. vertex shader

   ii. rasterizer

   iii. pixel shader

c) Briefly explain the purpose of any three coordinate systems (spaces) that you may encounter in a rendering pipeline. (8 marks)

QUESTION THREE (20 MARKS)

a) Define clipping and explain the various types of Text Clipping (8 marks)

b) Explain orthographic projections by using an example of a diagram of house (7 marks)

c) Consider two raster systems with the resolutions of 640 x 480 and 1280 x 1024.

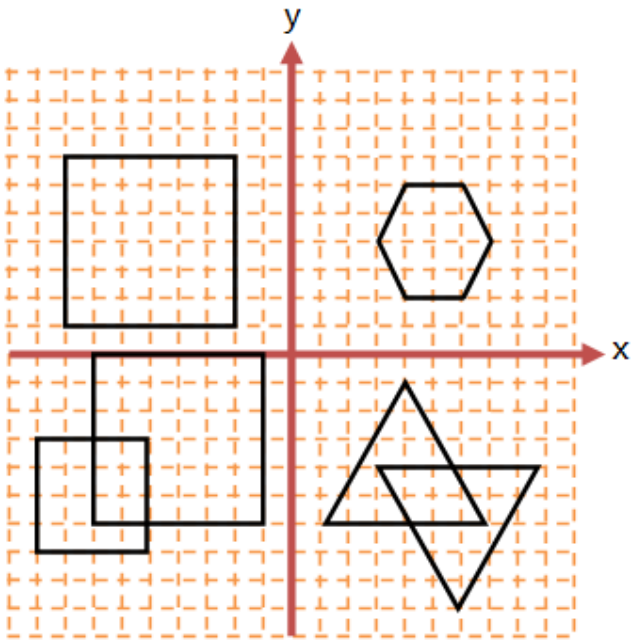
i. How many pixels could be accessed per second in each of these systems by a display controller that refreshes the screen at a rate of 60 frames per second (3 marks)

ii. What is the access time per pixel in each system (2 marks)

QUESTION FOUR (20 MARKS)

a) Explain any two methods used for smoothly joining two line segments (4 marks)

b) Write OpenGl sectional code to generate the output below (10 marks)



c) Using a diagram, explain window-to-viewport mapping (6 marks)

QUESTION FIVE (20 MARKS)

a) List out any two merits and demerits of Plasma panel display (4 marks)

b) Define Bresenham algorithm and develop the Bresenham's line drawing to draw lines of any scope. (8 marks)

c) Explain any four 2D Computer Graphics that are closely related to the six output functions of. Graphical Kernel System (GKS) (8 marks)