1. **SINGLE PROGRAM INTEGRATION**

**#include<stdio.h>**

**#include<process.h>**

**#include<window**

**#include<GL/glut.h**

**#include<math.h>**

**#include”program\_name.c”**

/\*all the eleven programs have to be included by writing their saved names\*/

/\* a function called write is defined, used to display the text on screen\*/

/\*glutBitmapCharacter renders a bitmap character using OpenGL.\*/

**void Write(char \*string){**

**while(\*string) {glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24,\*string++);**

**}**

**}**

/\*Without using any display lists, glutBitmapCharacter renders the character in the named bitmap font. \*/

**void OnMouseClick(int btn,int state,int x,int y){**

**if(btn==GLUT\_LEFT\_BUTTON&&state==GLUT\_DOWN){**

**printf("\n%d,%d",x,y);**

**if(x>=50&&x<=180&&y>=50&&y<=120){**

**spawnl(P\_NOWAIT, "tetra2.exe","tetra2.exe", NULL );**

**return;**

**}**

**if(x>=50&&x<=180&&y>=200&&y<=260){**

**Exit();**

**}**

**}**

**}**

/\*if the user clicks within these boundaries display function for tetrahedron i.e display is called,

/\*hence defining an if condition with boundaries of the button serve as implementation of a button

**void display()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT); //clears the color buffer**

**glColor3f (0.45,0.0,0.45); //set the color**

**glBegin(GL\_POLYGON); //to draw the polygon**

**glVertex2f(-0.4,-1.0);**

**glVertex2f(-1.0,-1.0);**

**glVertex2f(-1.0,1.0);**

**glVertex2f(-0.4,1.0);**

**glEnd(); //to end drawing the polygon**

**glBegin(GL\_POLYGON); //to draw the polygon**

**glColor3f (0.7,0.65,0.8); //set the color**

**glVertex2f(-0.4,1.0);**

**glColor3f(0.95,0.89,1); //set the color**

**glVertex2f(1.0,1.0);**

**glColor3f (0.7,0.65,0.8); //set the color**

**glVertex2f(1.0,-1.0);**

**glColor3f(0.53,0.46,0.67); //set the color**

**glVertex2f(-0.4,-1.0);**

**glEnd();**

**glColor3f (0.45,0.0,0.45);**

**glBegin(GL\_LINE\_LOOP); //to draw the border**

**glVertex2f(-0.3,0.9);**

**glVertex2f(0.9,0.9);**

**glVertex2f(0.9,-0.9);**

**glVertex2f(-0.3,-0.9);**

**glEnd();**

**glBegin(GL\_LINE\_LOOP); //to draw the border**

**glVertex2f(-0.28,0.87);**

**glVertex2f(0.88,0.87);**

**glVertex2f(0.88,-0.87);**

**glVertex2f(-0.28,-0.87);**

**glEnd();**

**glColor3f(0.3,0.1,0.4); //set the color**

**glRasterPos2f(-0.255,0.67); //to provide position for rasterization**

**Write1("COMPUTER GRAPHICS PROJECT WITH OPENGL");**

**glBegin(GL\_LINES); //to underline**

**glVertex2f(-0.255,0.663);**

**glVertex2f(0.855,0.663);**

**glEnd();**

**glColor3f(0,0.2,0.4); //set the color**

**glRasterPos2f(0.02,0.47); //to provide position for rasterization**

**Write1("A MENU BASED PACKAGE");**

**glBegin(GL\_LINES); //to underline**

**glVertex2f(0.02,0.463);**

**glVertex2f(0.64,0.463);**

**glEnd();**

**glColor3f(0,0.2,0.7); //set the color**

**glRasterPos2f(0.08,0.17); //to provide position for rasterization**

**Write2(" BY:");**

**glColor3f(0.3,0.2,0.4); //set the color**

**glRasterPos2f(0.08,-0.07); //to provide position for rasterization**

**Write1("NAME :AARSI");**

**glRasterPos2f(0.08,-0.37);**

**Write1("REG NO:14GAEI6001");**

**glRasterPos2f(0.08,-0.47);**

**Write1("CLASS:V SEM ISE");**

**glColor3f(0.8,0.1,0.2); //set the color**

**glRasterPos2f(0.08,-0.67); //to provide position for rasterization**

**Write2(" GUIDE:");**

**glRasterPos2f(0.08,-0.77); //to provide position for rasterization**

**Write1("Mrs.VIMALA H S");**

**glRasterPos2f(-0.23,-0.87); //to provide position for rasterization**

**GLUquadricObj \* quadricObj; //Creates a quadratic object**

**quadricObj = gluNewQuadric(); //Sets a pointer to a new quadratic object**

**gluQuadricDrawStyle(quadricObj, GLU\_FILL);**

**glPushMatrix();**

**glColor3f (0.4,0.3,0.8); //Sets ellipse color**

**glTranslatef(-0.7,0.91,0.0); //Translates the ellipse**

**glScalef(0.35, 0.089, 0.0); //Scales the ellipse**

**gluDisk(quadricObj, 0.0,0.75, 100, 100);**

**glPopMatrix();**

**glColor3f(1.0,1.0,1.0); //Sets ellipse color**

**glRasterPos2f(-0.89,0.88); //to provide position for rasterization**

**Write1("TETRAHEDRON");**

**glutSwapBuffers();**

**glFlush();**

**}**

**int main(int argc,char \*\*argv) //main function**

**{**

**glutInit(&argc,argv); //initializes GLUT**

**glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB); //defines the display mode**

**glutInitWindowSize(1000,600); //choose the window size**

**glutInitWindowPosition(100,100); //establish the window position**

**glutCreateWindow("PROJECT MENU CHART"); //create the window**

**glutMouseFunc(OnMouseClick); //for mouse callback**

**glutDisplayFunc(display); //for display callback**

**glutMainLoop();**

**}**