

PART-I: Lab for MCS-051 (Advanced Internet Technologies)

Q.1. Write a Program using Servlet and JDBC for developing online application for students attendance management for MCA –V semester students of a computer science department. Make necessary assumptions and create appropriate databases.

Q.2. Write a JSP Program, to manage saving account of a bank with basic feature including opening new account, cash deposit & withdrawal, providing account details and closing of accounts?

Q.3. Write an XML document to represent the medicines in a medical store. This document should include generic medicines, restricted medicines and life saving drugs.

Ans-:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<medical-medicenes>
< medical-medicenes >
< medicines >Generic medicines </ medicines>
<Generic Dieses>Genetic Patient</ Generic Dieses >
< medical 1>National Medical Store </ medical 1>
< medical 2>Nabeela Medical Store </ medical 2>
< /medical-medicenes >
< medical-medicenes >
< medicines > restricted medicines </ medicines>
< restricted Dieses>Poisson patient </ restricted Dieses >
< medical 1>Lucknow medical store</ medical 1>
< medical 2>Chowk medical store</ medical 2>
< /medical-medicenes >
< medical-medicenes >
< medicines> life saving drugs </ medicines >
< life saving medicineslife saving medicines >
< medical 1>Unique medical store</ medical 1>
< medical 2>Vilal medical store</ medical 2>
< /medical-medicenes >
< medical-medicenes >
```

PART-II: Lab for MCS-053 (Computer Graphics and Multimedia)

Q.1. a) Write a program in C/C++ using OpenGL to perform a 3-Dimensional transformation, such as translation, rotation and reflection, on a given triangle

Ans-: #include <GL/glut.h> // GLUT stuff, includes OpenGL headers as well
#include <windows.h>
#include <math.h>

// Point class to keep it a little cleaner.

```
class Point {  
public:  
    float x, y, z;  
    void setxy(float x2, float y2, float z2) { x = x2; y = y2; z=z2; }  
    const Point & operator=(const Point &rPoint) {  
        x = rPoint.x;  
        y = rPoint.y;  
        z = rPoint.z;  
        return *this;  
    }  
};  
  
void drawTriangle(Point p1, Point p2, Point p3) {  
    glBegin(GL_TRIANGLES);  
    glVertex3f(p1.x,p1.y,p1.z);//left of window  
    glVertex3f(p2.x,p2.y,p2.z);  
    glVertex3f(p3.x,p3.y,p3.z);  
    glEnd();//end drawing of line loop  
    glFlush();  
}  
  
void myDisplay() {  
    Point abc[3];  
    glMatrixMode(GL_MODELVIEW);  
    glClearColor(0.0,0.0,0.0,0.0);  
    glClear(GL_COLOR_BUFFER_BIT);  
    glLoadIdentity();  
    abc[0].x=1.0,abc[0].y=-0.25,abc[0].z=0.0;  
    abc[1].x=0.0,abc[1].y=-0.25,abc[1].z=0.0;  
    abc[2].x=0.0,abc[2].y=0.25,abc[2].z=0.0;  
    drawTriangle(abc[0],abc[1],abc[2]);  
    glTranslatef(0.0f,-0.6f,0.0f);  
    drawTriangle(abc[0],abc[1],abc[2]);  
    glRotatef(180.0,0.0f,0.0f,1.0f);  
    drawTriangle(abc[0],abc[1],abc[2]);  
}  
  
int main(int argc, char *argv[]) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);  
    glutInitWindowSize(640,480);  
    glutInitWindowPosition(100,150);  
    glutCreateWindow("3D Transformation");  
    glutDisplayFunc(myDisplay);  
    glutMainLoop();  
    return 0;  
}
```

Nabeela khaan

IVC

b) Write a program in C/C++ using OpenGL to implement the Cohen Sutherland line clipping algorithm.

Q.2. Write a program in C/C++ using OpenGL to draw a polygon having 4 vertices A (4, 2) B (8, 2), C (4, 12), D (8, 12), it is reflected about the line $y = 2$. Using homogeneous coordinate system this program should find the coordinates of the reflected vertices.

Ans-:

Q.3. Write a program in C/C++ using OpenGL to draw Bezier curves, make necessary assumption and take required input from console.

Ans:- Let, a Bezier curve having the control points as $p_0 (0, 0)$, $P_1 (2, 5)$, $P_2 (5, 9)$, $P_3 (10, 20)$. Calculate the coordinates of the points on the curve corresponding to the parameter $u = 0.2, 0.4, 0.6$.

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

// Point class to keep it a little cleaner.
class Point {
public:
    float x, y;
    void setxy(float x2, float y2) { x = x2; y = y2; }
    const Point & operator=(const Point &rPoint) {
        x = rPoint.x;
        y = rPoint.y;
        return *this;
    }
};

Point abc[4];

void myInit() {
    glClearColor(0.0,0.0,0.0,0.0);
    glColor3f(1.0,0.0,0.0);
    glPointSize(8.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,64.0,0.0,48.0);
}

void drawDot(int x, int y) {
    glBegin(GL_POINTS);
    glVertex2i(x,y);
    glEnd();
    glFlush();
}
```

```
void drawLine(Point p1, Point p2) {  
    glBegin(GL_LINES);  
    glVertex2f(p1.x, p1.y);  
    glVertex2f(p2.x, p2.y);  
    glEnd();  
    glFlush();  
}  
  
// Calculate the next bezier point.  
Point drawBezier(Point A, Point B, Point C, Point D, double t) {  
    Point P;  
    P.x = pow((1 - t), 3) * A.x + 3 * t * pow((1 - t), 2) * B.x + 3 * (1 - t) * pow(t, 2) * C.x + pow(t, 3) * D.x;  
    P.y = pow((1 - t), 3) * A.y + 3 * t * pow((1 - t), 2) * B.y + 3 * (1 - t) * pow(t, 2) * C.y + pow(t, 3) * D.y;  
    return P;  
}  
  
void myDisplay() {  
    glClear(GL_COLOR_BUFFER_BIT);  
    Point POld=abc[0];  
    abc[0].x=0,abc[0].y=0;  
    abc[1].x=2,abc[1].y=5;  
    abc[2].x=5,abc[2].y=9;  
    abc[3].x=10,abc[3].y=20;  
    glColor3f(1.0,0.0,0.0);  
    for(int i=0;i<4;i++)  
        drawDot(abc[i].x, abc[i].y);  
    glColor3f(1.0,1.0,1.0);  
    drawLine(abc[0], abc[1]);  
    drawLine(abc[1], abc[2]);  
    drawLine(abc[2], abc[3]);  
    for(double t = 0.0;t <= 1.0; t += 0.1) {  
        Point P = drawBezier(abc[0], abc[1], abc[2], abc[3], t);  
        drawLine(POld, P);  
        POld = P;  
    }  
    glFlush();  
}  
  
int main(int argc, char *argv[]) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);  
    glutInitWindowSize(640,480);  
    glutInitWindowPosition(100,150);  
    glutCreateWindow("Bezier Curve");  
    glutDisplayFunc(myDisplay);  
    myInit();  
    glutMainLoop();  
    return 0;  
}
```

Output



Solved By Nabeela khaan