

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

Q1:

Write a Program using Servlet and JDBC for developing online application for displaying TEE results of MCA Program. A student has to score 50 % in theory, practical and assignment to qualify the paper. Create appropriate databases.

Database creation

```
CREATE TABLE Student
```

```
(
    student_id INT NOT NULL AUTO_INCREMENT,
    student_name VARCHAR(25),
    student_enrol VARCHAR(10),
    student_add VARCHAR(70),
    student_prog VARCHAR(10),
    student_prog_start VARCHAR(5),
    student_prog_end VARCHAR(5),
    PRIMARY KEY(student_id)
);
```

```
CREATE TABLE Course (
    course_id INT NOT NULL AUTO_INCREMENT,
    course_code VARCHAR(10),
    course_name VARCHAR(20),
    course_prog VARCHAR(10),
    PRIMARY KEY(course_id)
);
```

```
CREATE TABLE Result (
    ID INT NOT NULL AUTO_INCREMENT,
    course_code VARCHAR(10),
    student_id VARCHAR(10),
    marks_assign_got VARCHAR(5),
    marks_pract_got VARCHAR(5),
    marks_theory_got VARCHAR(5),
    result_status VARCHAR(10),
    PRIMARY KEY(ID),
    FOREIGN
    KEY(course_code, student_id, marks_assign_got, marks_pract_got, marks_theory_got)
    REFERENCES Course (course_code)
    , Student(student_id), Marks(marks_assign_got, marks_pract_got, marks_theory_got)
);
```

Result Servlet

CODE

```
package ignou;
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.net.URL;
public class Result extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
    ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
```

```

out.println("<HTML><HEAD><TITLE>IGNOU MCA TEE Result:::</TITLE>");
out.println("</HEAD>");
out.println("<BODY >");
    out.println("<P align=center>MCA TEE Result</P>");
    out.println("<TABLE align=center border=1 width=\"65%\">");
Connection con=null;
Statement stm = null;
try{
    Class.forName("org.gjt.mm.mysql.Driver");
    String dbURL="jdbc:mysql://localhost/MCA";
    String username="root";
    String password="";
    con=DriverManager.getConnection(dbURL,username,password);
    stm=con.createStatement();
    ResultSet rs=stm.executeQuery("SELECT
student_name,course_code,marks_assign_got,marks_theory_got,marks_pract_got,result_status FROM Student,Result WHERE Result.student_id =Student.student_id");
    while(rs.next()) //retrieval all the records into the table
    {
        out.println("<TR>");
        out.println("<TD>Name</TD><TD>" + rs.getString("student_name") +
"</TD></TR>");
        out.println("<TR><TD>Course code</TD><TD>" + rs.getString("course_code")
+ "</TD></TR>");
        out.println("<TR><TD>Assignments marks</TD><TD>" +
rs.getString("marks_assign_got") + "</TD></TR>");
        out.println("<TR><TD>Theory marks</TD><TD>" +
rs.getString("marks_theory_got") + "</TD></TR>");
        out.println("<TR><TD>Practical marks</TD><TD>" +
rs.getString("marks_pract_got") + "</TD></TR>");
        out.println("<TR><TD>Result</TD><TD>" + rs.getString("result_status") +
"</TD>");
        out.println("</TR>");
        rs.close();
        stm.close();
        con.close();
    }}catch(Exception e){
    out.println(e.getMessage());
}
    out.println("<P>&nbsp;</P></FONT></BODY></HTML>");
}
}

```

Q2:

Write a JSP Program, which displays a web page containing the name of the school, program being offered currently, number of students enrolled in each program, new programs to be offered, eligibility criteria for taking admission in each program.

course.jsp

CODE

```

<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
    <title>::Our available courses/program::</title>
<style>
BODY{scrollbar-face-color:#708090;
scrollbar-arrow-color:#FFFFFF;

```

```

scrollbar-track-color:#DDDDFF;
scrollbar-shadow-color:'';
scrollbar-highlight-color:'';
scrollbar-3dlight-color:'';
scrollbar-darkshadow-color:''
}/>
</head>

<body txt color="#000000">

<table cellpadding="5" border="0" width="756">

<!-- start the HTML for the header row -->

<h3 align="center"><font color="#ee55ee">Welcome To NICS(National Institute Of
Computer Science)</h3>

    <td width="404" valign="top" >
        <h3 align="center">Available Degree and Master degree Courses</h3>

        <p>
        <ul>
        <li type=square>BCA[Eligibility:H.S in any Stream having maths in
        12]<br>Registered students:50</li>

        <li type=square>MCA[Eligibility:Any degree having maths in 12]<br>Registered
        students:70</li>

        </td>

        <td width="160" valign="top" >
            <p>Our upcoming courses .
            <p>
            <ul>
            <li type=circle>BIT</li>
            <li type=circle>Bsc IT</li>
            <li type=circle>MBA(IT)</li>
            <li type=circle>MTech(IT)</li>

            </td>
        </tr>
        <tr>
            <td colspan="3"><center>
                <p class="copyright"><font color="#000000">
                    &copy; NICS&nbsp;Copyright 2008.
                    All rights reserved.</f></p>
            </td>
        </tr>
    </table>
</body>
</html>

```

Q3:

Write a program using JDBC and JSP to display the names and addresses of all those MCA students who are working in Software Development Company.

CODE

DATABASE

Database name: Company

```
CREATE TABLE Student (  
  StudentID VARCHAR(10) NOT NULL ,  
  Student_name VARCHAR(30),  
  Student_enrol VARCHAR(9),  
  Student_address VARCHAR(70),  
  Student_course VARCHAR(10),  
  CentreID VARCHAR(4),  
  PRIMARY KEY(StudentID)  
);
```

```
CREATE TABLE Soft_company (  
  comp_id INT NOT NULL AUTO_INCREMENT,  
  comp_name VARCHAR(30),  
  comp_address VARCHAR(70),  
  StudentID VARCHAR(10),
```

```
  PRIMARY KEY(comp_id),  
  FOREIGN KEY(StudentID) REFERENCES Student (StudentID)  
);
```

JSP page for displaying students who are working in soft dev company

CODE

```
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">  
<html>  
<head>  
<title>Working MCA students</title>  
</head>  
  
<body bgcolor=#ffccdd text=#111ccc>  
  
<h3>Information of students who are currently working in software development  
Company</h3>  
  
<table cellpadding="5" cellspacing="5" border="1" align="center" width="70%">  
<tr>  
<td align="right">Name:</td>  
  
<td align="right">Address:</td>  
  
<td align="right">Software Company:</td>  
</tr>  
<%@page language="java"%>  
<%@page import="java.sql.*"%>  
<%  
  
Connection con = null;  
  
Statement stm = null;  
  
try{  
  Class.forName("org.gjt.mm.mysql.Driver");  
  String dbURL="jdbc:mysql://localhost/Company";
```

```
String username="root";
String password="";

con=DriverManager.getConnection(dbURL,username,password);
stm=con.createStatement();

ResultSet rs=stm.executeQuery("SELECT
Student.Student_name,Student.Student_address,Soft_company.comp_name FROM
Student,Soft_company WHERE Student.StudentID=Soft_company.StudentID");

while(rs.next()){%>
<TR>
<TD><%out.println(rs.getString("Student_name")); %></TD>
<TD><%out.println(rs.getString("Student_address")); %></TD>
<TD><%out.println(rs.getString("comp_name")); %></TD>

</TR>
<%}%>
<%

rs.close();
stm.close();
con.close();

} catch(Exception e){
e.printStackTrace();
}

%>

</table>
</body>
</html>
```

Q4:

Write an XML document to represent the TEE grade card of MCA that will contain student name, subjects, assignment marks, total marks, and final status (pass/fail).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Gradecard SYSTEM"c:/xml/gradecard.dtd">
<!--Award list MCA final year student-->
<Gradecard>
<students>
<student contact="yes/no">
<name>student_name</name>
<subjects>
<subject>
<subjectCode>subject_code</subjectCode>
<assignMarks>assignment_marks</assignMarks>
<totalMarks>total_including_pract_theory</totalMarks>
<status>pass/fail</status>
</subject>
</subjects>
</student>
</students>
```

</Gradecard>

DTD for above xml file

```
<!DOCTYPE Gradecard [  
<!ELEMENT students (student+)>  
<!ELEMENT student(name,subjects)>  
<!ATTLIST student contact(yes|no)"yes">  
<!ELEMENT name(#PCDATA)>  
<!ELEMENT subjects(subject)>  
<!ELEMENT subject(subjectCode,assignMarks,totalMarks,status)>  
<!ELEMENT subjectCode(#PCDATA)>  
<!ELEMENT assignMarks(#PCDATA)>  
<!ELEMENT totalMarks (#PCDATA)>  
<!ELEMENT status (#PCDATA)>  

```

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

Q1:

Write a program in C or C++ demonstrate Bresenham's Line generation algorithm?

```
/*Line generation using Bresenham's algo*/
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<graphics.h>
```

```
void line_bre(int xa,int ya,int xb,int yb)
```

```
{int dx=xb-xa;
```

```
int dy=yb-ya;
```

```
int steps,k,p;
```

```
float x,y;
```

```
x=xa;
```

```
y=ya;
```

```
if(abs(dx)>abs(dy))
```

```
{
```

```
    steps=abs(dx);
```

```
    p=2*dy-dx;
```

```
}
```

```
else
```

```
{
```

```
    steps=abs(dy);
```

```
    p=2*dx-dy;
```

```
}
```

```
putpixel(x,y,4);
```

```
for(k=0;k<steps;k++)
```

```
{
```

```
    if(abs(dx)>abs(dy))
```

```
    {
```

```
        if(p<0)
        {
            x=x+1;
            p=p+2*dy;
        }
        else
        {
            x=x+1;
            y=y+1;
            p=p+2*dy-2*dx;
        }
    }
    else
    {
        if(p<0)
        {
            y=y+1;
            p=p+2*dx;
        }
        else
        {
            x=x+1;
            y=y+1;
            p=p+2*dx-2*dy;
        }
    }
    putpixel(x,y,4);
}
void main()
{
    int gd=DETECT,gm,x1,y1,x2,y2;
    clrscr();
    initgraph(&gd, &gm, "");
    printf("\nEnter coordinates for a line (x1,y1,x2,y2)\n ");
    scanf("%d %d %d %d",&x1,&y1,&x2,&y2);
    clrscr();
    line_bre(x1,y1,x2,y2);
    getch();
}
```

Q2:

Using OpenGL write a program to develop a chessboard pattern on the screen?

```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
#include <stdlib.h>
#include <stdio.h>
```

```
/* Create checkerboard texture */
#define checkImageWidth 64
#define checkImageHeight 64
static GLubyte checkImage[checkImageHeight][checkImageWidth][4];

static GLuint texName;

void makeCheckImage(void)
{
    int i, j, c;

    for (i = 0; i < checkImageHeight; i++) {
        for (j = 0; j < checkImageWidth; j++) {
            c = (((i&0x8)==0)^((j&0x8)==0))*255;
            checkImage[i][j][0] = (GLubyte) c;
            checkImage[i][j][1] = (GLubyte) c;
            checkImage[i][j][2] = (GLubyte) c;
            checkImage[i][j][3] = (GLubyte) 255;
        }
    }
}

void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);

    makeCheckImage();
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);

    glGenTextures(1, &texName);
    glBindTexture(GL_TEXTURE_2D, texName);

    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
        GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
        GL_NEAREST);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth,
        checkImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE,
        checkImage);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_TEXTURE_2D);
```



```
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
glBindTexture(GL_TEXTURE_2D, texName);
glBegin(GL_QUADS);
glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, -1.0, 0.0);
glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 1.0, 0.0);
glTexCoord2f(1.0, 1.0); glVertex3f(0.0, 1.0, 0.0);
glTexCoord2f(1.0, 0.0); glVertex3f(0.0, -1.0, 0.0);
```

```
glTexCoord2f(0.0, 0.0); glVertex3f(1.0, -1.0, 0.0);
glTexCoord2f(0.0, 1.0); glVertex3f(1.0, 1.0, 0.0);
glTexCoord2f(1.0, 1.0); glVertex3f(2.41421, 1.0, -1.41421);
glTexCoord2f(1.0, 0.0); glVertex3f(2.41421, -1.0, -1.41421);
glEnd();
glFlush();
glDisable(GL_TEXTURE_2D);
}
```

```
void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 30.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef(0.0, 0.0, -3.6);
}
```

```
void keyboard (unsigned char key, int x, int y)
{
    switch (key) {
    case 27:
        exit(0);
        break;
    default:
        break;
    }
}
```

```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(250, 250);
    glutInitWindowPosition(100, 100);
    glutCreateWindow(argv[0]);
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
}
```

```
glutKeyboardFunc(keyboard);  
glutMainLoop();  
return 0;  
}
```

Part-II:Q2

The following code supposedly generates a chessboard pattern. To run the executables, you may need the file `opengl32.dll`, and to compile the source files, you will need the GL, GLU, and GLUT header files.

CODE

```
PImage textureImg =  
loadImage("vetro.jpg"); // dummy image colorMode(RGB,1);  
  
int biro = 0;  
int bbiro = 0;  
int scacco = 5;  
for (int i=0; i<textureImg.width; i+=scacco) {  
    bbiro = (bbiro + 1)%2; biro = bbiro;  
    for (int j=0; j<textureImg.height; j+=scacco) {  
        for (int r=0; r<scacco; r++)  
            for (int s=0; s<scacco; s++)  
                textureImg.set(i+r,j+s, color(biro));  
        biro = (biro + 1)%2;  
    }  
}  
image(textureImg, 0, 0);
```

Q3:

Using OpenGL develop a scene of rising and setting Sun?

```
#include <GL/gl.h>  
#include <GL/glu.h>  
#include <GL/glut.h>  
#include <stdlib.h>  
  
static GLfloat xspin = 0.0, yspin=0.0, hasCrossed=0.0;  
static GLfloat fcol = 0.0, scol=0.0, tcol=0.0, focol=0.0, colIncr=0.0005, yIncr=0.0;  
static GLfloat sunf=1.0, suns=1.0, sunt=0.0;  
void init(void)  
{  
  
}  
void display(void)  
{  
    glClearColor (fcol, scol, tcol, focol);  
    glShadeModel (GL_FLAT);  
    glClear(GL_COLOR_BUFFER_BIT);  
    glPushMatrix();  
    glTranslatef(xspin, yspin, 1.0);  
    glColor3f(sunf, suns, sunt);  
    glutSolidSphere(15, 100, 20);  
    glPopMatrix();  
}
```

```
glutSwapBuffers();
}
void spinDisplay(void)
{

xspin = xspin + 0.025;
if (xspin > 100)
xspin = xspin - 100;

hasCrossed=hasCrossed+0.025;

if (hasCrossed < 50)
yspin = yspin + 0.025;
else
yspin = yspin - 0.025;

if(hasCrossed>100)
hasCrossed=0.0;

if (hasCrossed < 50)
{
fcol=fcol+colIncr;
scol=scol+colIncr;

}
else
{

fcol=fcol-colIncr;
scol=scol-colIncr;

}

if(fcol>1)
{
fcol=0.0;
scol=0.0;

}

glutPostRedisplay();

}
void reshape(int w, int h)
{
glViewport (0, 0, (GLsizei) w, (GLsizei) h);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-10, 100, -20, 100, -20, 20);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
}
void mouse(int button, int state, int x, int y)
{
switch (button) {
```

```
case GLUT_LEFT_BUTTON:
if (state == GLUT_DOWN)
glutIdleFunc(spinDisplay);
break;
case GLUT_MIDDLE_BUTTON:
if (state == GLUT_DOWN)
glutIdleFunc(NULL);
break;
default:
break;
}
}
/*
* Request double buffer display mode.
* Register mouse input callback functions
*/
int main(int argc, char** argv)
{
glutInit(&argc, argv);
glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
glutInitWindowSize (500, 500);
glutInitWindowPosition (100, 100);
glutCreateWindow (argv[0]);
init ();
glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutMouseFunc(mouse);
glutMainLoop();
return 0;
}
```

Q4:

Write a program in C or C++ to produce the sweep representation of a circle and hence produce a cylinder?

/* A C++ Program that creates a 3D solid object using Translational Sweep Representation Method.*/

//----- Header Files -----//

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <ctype.h>
#include <stdlib.h>
#include <stdio.h>
```

//----- Global Declarations -----//

```
# define f 0.3
# define projection_angle 45
```

//----- Function Prototypes -----//

```
void show_screen( );
```

```
void apply_translation(const int,const int,const int,
const int,const int,const int);
void multiply_matrices(const float[4],const float[4][4],float[4]);

void draw_circle(const int,const int,const int);
void get_projected_point(int&,int&,int&);

void Line(const int,const int,const int,const int);

//----- main( ) -----//

int main( )
{
int driver=VGA;
int mode=VGAHI;

initgraph(&driver,&mode,"..\\Bgi");

show_screen( );

int x=0;
int y=320;
int z=0;

setcolor(15);
draw_circle(x,y,z);

setcolor(15);
settextstyle(0,0,1);
outtextxy(50,415,"*** Press any key to make a 3D solid object i.e. Cylinder.");

getch( );

for(int t=1;t<=150;t++)
apply_translation(x,y,z,0,-t,0);

getch( );
return 0;
}

//----- Funcion Definitions -----//

//----- apply_translation( ) -----//

void apply_translation(const int x,const int y,const int z,
const int Tx,const int Ty,const int Tz)
{
int _x=x;
int _y=y;
int _z=z;

float matrix_a[4]={_x,_y,_z,1};

float matrix_b[4][4]={
```

```
{ 1,0,0,0 } ,
{ 0,1,0,0 } ,
{ 0,0,1,0 } ,
{ Tx,Ty,Tz,1 }
};

float matrix_c[4]={0};

multiply_matrices(matrix_a,matrix_b,matrix_c);

_x=(int)(matrix_c[0]+0.5);
_y=(int)(matrix_c[1]+0.5);
_z=(int)(matrix_c[2]+0.5);

draw_circle(_x,_y,_z);
}

//----- multiply_matrices( ) -----//

void multiply_matrices(const float matrix_1[4],
const float matrix_2[4][4],float matrix_3[4])
{
for(int count_1=0;count_1<4;count_1++)
{
for(int count_2=0;count_2<4;count_2++)
matrix_3[count_1]+=
(matrix_1[count_2]*matrix_2[count_2][count_1]);
}
}

//----- draw_circle( ) -----//

void draw_circle(const int x,const int y,const int z)
{
int _x=x;
int _y=y;
int _z=z;

get_projected_point(_x,_y,_z);

_x+=320;

setcolor(15);
setfillstyle(1,0);
fillcircle(_x,_y,75,50);
}

//----- get_projected_point( ) -----//

void get_projected_point(int& x,int& y,int& z)
{
float fcos0=(f*cos(projection_angle*(M_PI/180)));
float fsin0=(f*sin(projection_angle*(M_PI/180)));

float Par_v[4][4]={
```

```
{1,0,0,0},
{0,1,0,0},
{fcos0,fsin0,0,0},
{0,0,0,1}
};

float xy[4]={x,y,z,1};
float new_xy[4]={0};

multiply_matrices(xy,Par_v,new_xy);

x=(int)(new_xy[0]+0.5);
y=(int)(new_xy[1]+0.5);
z=(int)(new_xy[2]+0.5);
}

//----- show_screen( ) -----//

void show_screen( )
{
setfillstyle(1,1);
bar(140,26,485,38);

settextstyle(0,0,1);
setcolor(15);
outtextxy(5,5,"*****");
outtextxy(5,17,"*-
*****-*");
outtextxy(5,29,"*------*");
outtextxy(5,41,"*-
*****-*");
outtextxy(5,53,"*-
*****-*");
setcolor(11);
outtextxy(145,29,"Translational Sweep Representation Method");
setcolor(15);
for(int count=0;count<=30;count++)
outtextxy(5,(65+(count*12)),"*- *-*");
outtextxy(5,438,"*-
*****-*");
outtextxy(5,450,"*------*");
outtextxy(5,462,"*****");
setcolor(12);
outtextxy(229,450,"Press any Key to exit.");
}
//----- THE END -----//
```