CS 352

Computer Graphics & Visualization

Assignment - 2

Name – Anjani Kumar

Roll No - 210001004

Question – 1:

CODE:

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

#include<bits/stdc++.h> // Standard C++ library headers

using namespace std;


int x_coor, y_coor; // Variables to store the coordinates of the bottom-most

// left vertex of the rectangle

int length, breadth; // Variables to store the length and breadth of the rectangle

int x_mov, y_mov; // Variables to store the movement in x-direction and y-
// direction after clicking on the key


// Display callback function

void displayCB()

{

glClear(GL_COLOR_BUFFER_BIT); // Clear the color buffer


// Draw a colored polygon

glColor3f(0.4, 0.6, 0.4); // Set color to a shade of green

glBegin(GL_POLYGON);

glVertex2f(x_coor, y_coor);

glVertex2f(x_coor + breadth, y_coor);

glVertex2f(x_coor + breadth, y_coor + length);

glVertex2f(x_coor, y_coor + length);

glEnd();
```

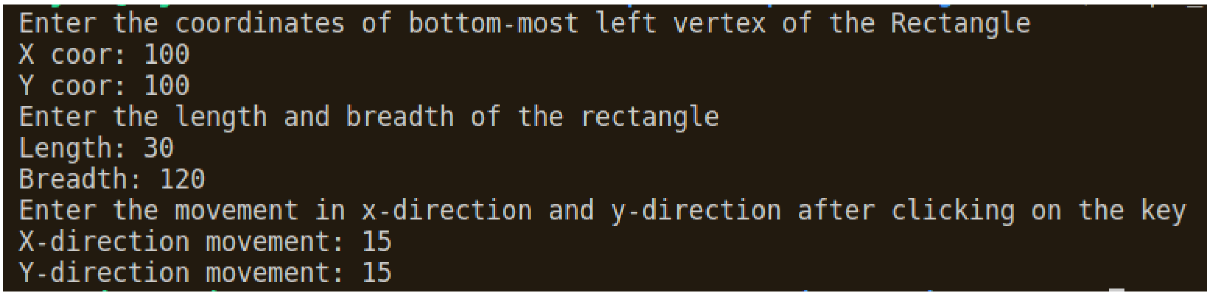
```
glFlush(); // Flush OpenGL buffers to display
}

// Function to handle special keyboard key events
void Keys(int key, int x, int y)
{
// Move the rectangle based on the keys pressed
if(key == 101){ // Up arrow key
y_coor += y_mov;
}
if(key == 100){ // Left arrow key
x_coor -= x_mov;
}
if(key == 103){ // Down arrow key
y_coor -= y_mov;
}
if(key == 102){ // Right arrow key
x_coor += x_mov;
}

// Ensure that the rectangle stays within the window boundaries
if(x_coor < 0) x_coor = 0;
if(x_coor + breadth > 600) x_coor = 600 - x_mov;
if(y_coor < 0) y_coor = 0;
if(y_coor + length > 600) y_coor = 600 - y_mov;

glutPostRedisplay(); // Request a redraw to update the display
```

```
}

// Main function
int main(int argc, char *argv[])
{
// Input the coordinates, length, breadth, and movement from the user
cout<<"Enter the coordinates of bottom-most left vertex of the Rectangle\n";
cout<<"X coor: ";
cin>>x_coor;
cout<<"Y coor: ";
cin>>y_coor;

cout<<"Enter the length and breadth of the rectangle"<<endl;
cout<<"Length: ";
cin>>length;
cout<<"Breadth: ";
cin>>breadth;
cout<<"Enter the movement in x-direction and y-direction after clicking on the
key\n";
cout<<"X-direction movement: ";
cin>>x_mov;
cout<<"Y-direction movement: ";
cin>>y_mov;

// Initialize GLUT
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB);
glutInitWindowSize(600, 600);
```

```
glutCreateWindow("Question_01 - Colored Polygon Shifting");

// Set up the OpenGL environment
glClearColor(1, 1, 1, 0.0); // Set clear color to white
gluOrtho2D(0, 600, 0, 600); // Set up a 2D orthographic projection

// Register the display callback function and the special keys callback function
glutDisplayFunc(displayCB);
glutSpecialFunc(Keys);

// Enter the GLUT event processing loop
glutMainLoop();

return 0; // Return 0 to exit main function
}
```

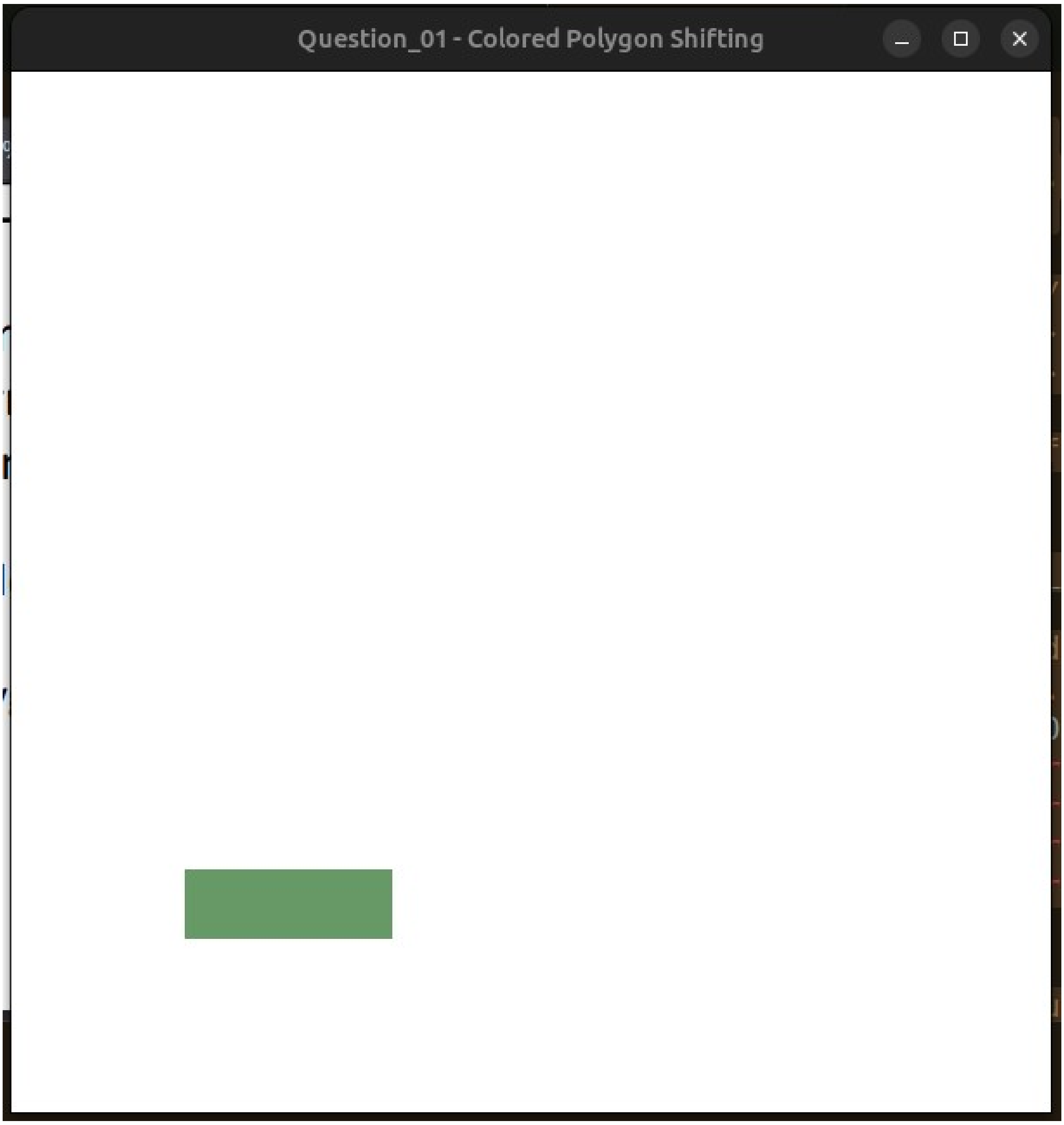
Screenshot of the Output:

a) Given Input:

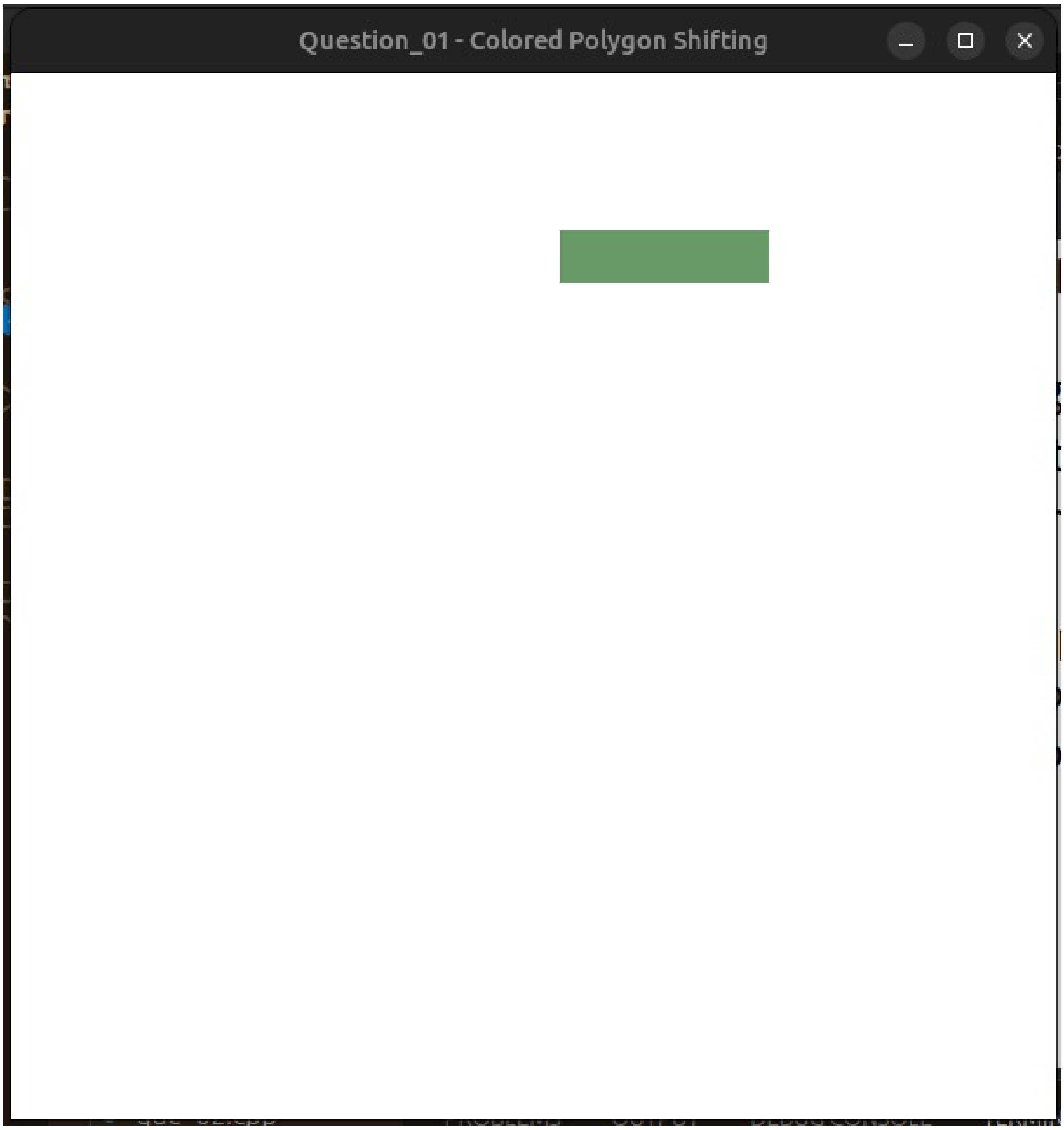


```
Enter the coordinates of bottom-most left vertex of the Rectangle
X coor: 100
Y coor: 100
Enter the length and breadth of the rectangle
Length: 30
Breadth: 120
Enter the movement in x-direction and y-direction after clicking on the key
X-direction movement: 15
Y-direction movement: 15
```

b) Initial Position:



c) After doing the operations



Question_01 - Colored Polygon Shifting

Question 2:

Code:

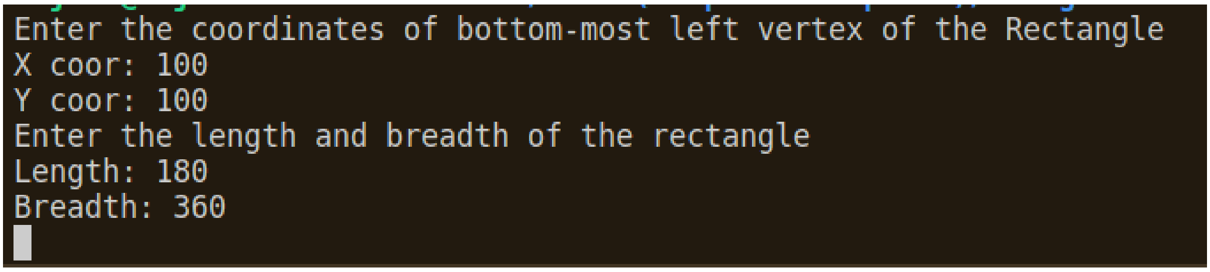
```
#include <GL/glut.h>
#include<bits/stdc++.h> // Standard C++ library headers
using namespace std;

// Global variables to store the coordinates of the bottom-most left vertex of the rectangle
int x_coor, y_coor;

// Global variables to store the length and breadth of the rectangle
int length, breadth;

// Global variables to store the RGB components of the color
float red = 1, blue = 0, green = 1;

// Display callback function
void displayCB(void)
{
glClear(GL_COLOR_BUFFER_BIT); // Clear the color buffer

// Draw a colored polygon with the specified color
glColor3f(red, green, blue);
glBegin(GL_POLYGON);
glVertex2f(x_coor, y_coor);
glVertex2f(x_coor + breadth, y_coor);
glVertex2f(x_coor + breadth, y_coor + length);
```

```
glVertex2f(x_coor, y_coor + length);
glEnd();

glFlush(); // Flush OpenGL buffers to display
}

// Mouse click callback function
void mouseClick(int button, int state, int x, int y)
{
if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
{
// Change color to a random one on left-click
red = (GLfloat)rand() / RAND_MAX;
green = (GLfloat)rand() / RAND_MAX;
blue = (GLfloat)rand() / RAND_MAX;
glutPostRedisplay(); // Request a redraw to update the display
}
}

int main(int argc, char *argv[])
{
// Input the coordinates, length, breadth, and movement from the user
cout << "Enter the coordinates of bottom-most left vertex of the Rectangle\n";
cout << "X coor: ";
cin >> x_coor;
cout << "Y coor: ";
cin >> y_coor;
```

```
cout << "Enter the length and breadth of the rectangle" << endl;
cout << "Length: ";
cin >> length;
cout << "Breadth: ";
cin >> breadth;

// Initialize GLUT
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB);
glutInitWindowSize(600, 600);
glutCreateWindow("Question_02 - Color Changing of the Polygon");

// Set up the OpenGL environment
glClearColor(1, 1, 1, 0.0); // Set clear color to white
gluOrtho2D(0, 600, 0, 600); // Set up a 2D orthographic projection

// Register the display callback function and the mouse click callback function
glutDisplayFunc(displayCB);
glutMouseFunc(mouseClick);

glutMainLoop();

return 0;
}
```

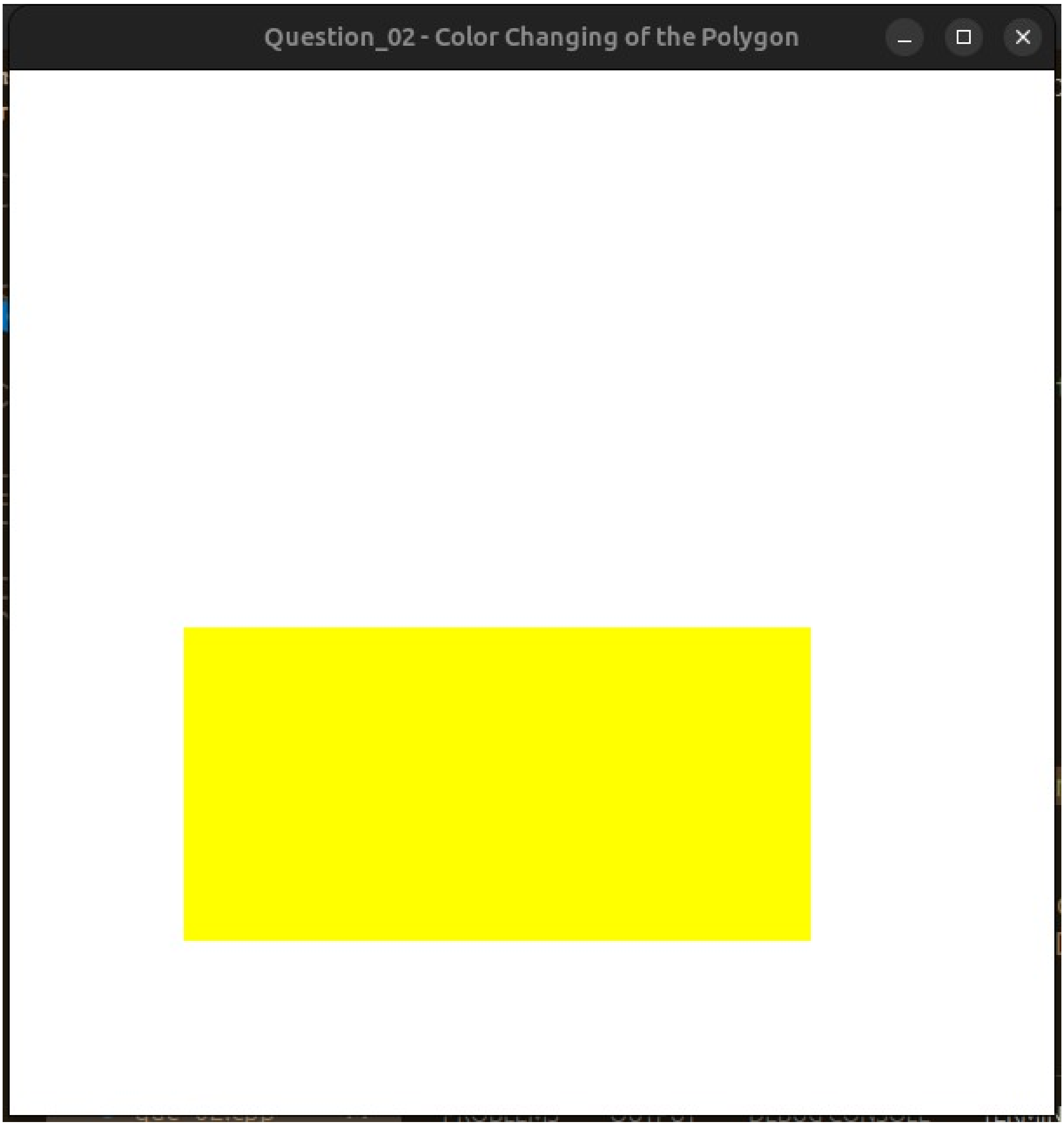
Screenshot of the ouput:

a) Given Input:

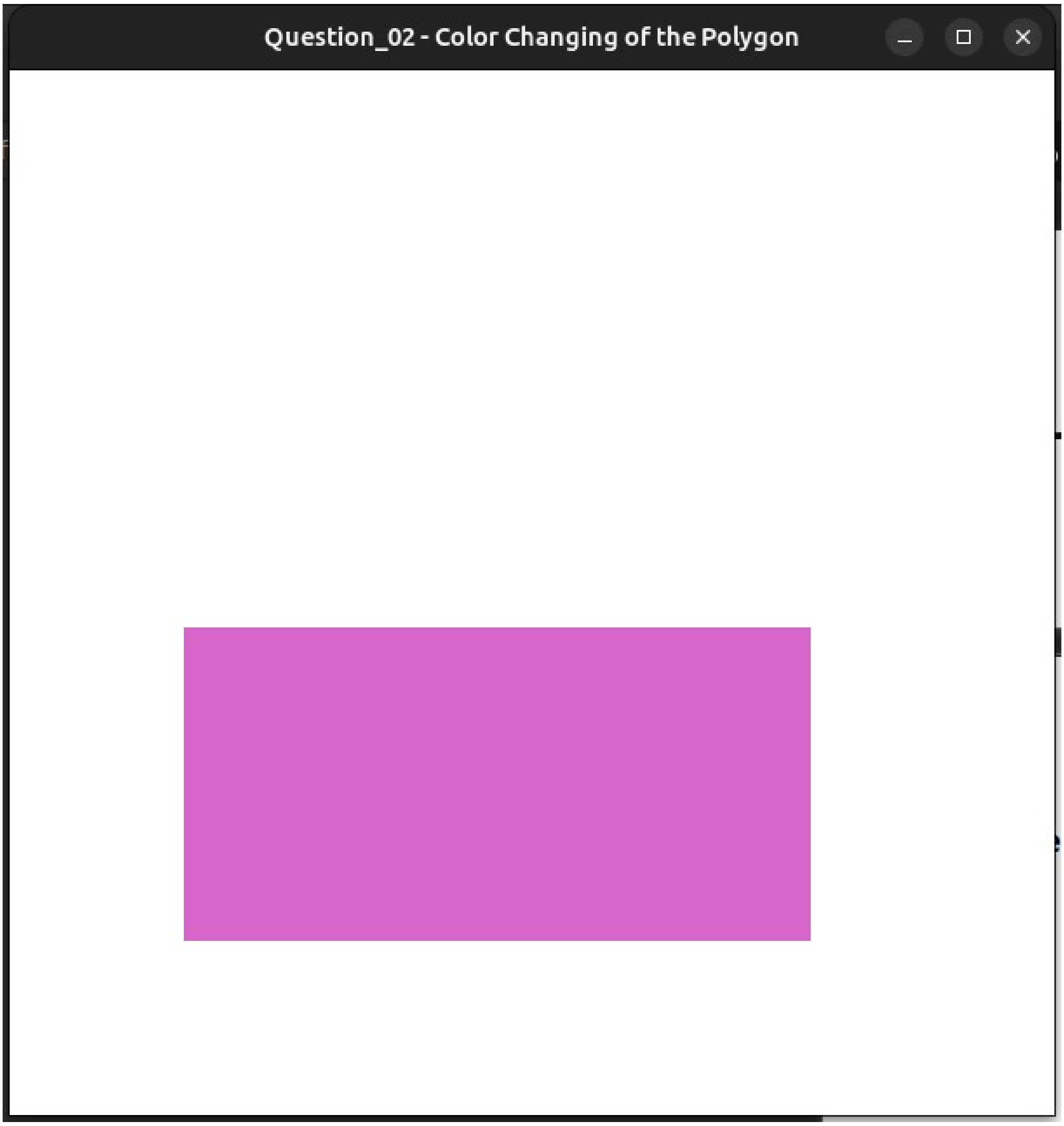


```
Enter the coordinates of bottom-most left vertex of the Rectangle
X coor: 100
Y coor: 100
Enter the length and breadth of the rectangle
Length: 180
Breadth: 360
```

b) Initial Condition of the Rectangle:



c) Rectangle after changing the color:



Question_02 - Color Changing of the Polygon