**Lab File**

**Computer Graphics**

**[CSE203]**

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



**Submitted To: Submitted By:**

 Dr. S. K. Dubey Kuldeep Dwivedi

Associate Professor A2305218477

CSE Department, ASET B. Tech (CSE)

6CSE7-Y

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

NOIDA-201301

**INDEX**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Experiment No.** | **Category**  **of Assignment** | **Code** | **Name of Experiment** | **Date of Allotment of experiment** | **Date of Evaluation** | **Max.**  **Marks** | **Marks obtained** | **Sign.**  **of Faculty** |
|  | **Mandatory Experiment**  **Mandatory Experiment** | **LR (0)** | Write a program to implement various Graphics Functions | **8/1/2021** | **15/1/2021** | **1** |  |  |
|  | Write a Program to draw a line using DDAAlgorithms | **15/1/2021** | **22/1/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a Program to draw a line using Bresenhem'sAlgorithm | **22/1/2021** | **29/1/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a Program to draw a circle using Bresenhem's Algorithm | **29/1/2021** | **5/2/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a program to draw a circle using mid-point algorithm | **5/2/2021** | **12/2/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a program to draw a ellipse using mid-point algorithm | **12/2/2021** | **19/2/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a program for translation and scaling of the triangle | **19/2/2021** | **5/3/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a program for rotation of the triangle | **5/3/2021** | **12/3/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a program for reflection of the triangle | **12/3/2021** | **19/3/2021** | **1** |  |  |
|  | **Mandatory Experiment** |  | Write a program for Cohen- Sutherland clipping algorithm | **19/3/2021** | **26/3/2021** | **1** |  |  |

# **EXPERIMENT 1**

**Date:21/12/2020**

**Objective:** Write a program to implement Computer Graphics Functions

**Software Used:** Turbo C++

**Theory:**

**Code:**

**//Program to draw a line:**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**line(100,150,560,670);**

**getch();**

**return 0;**

**}**

**//Program to draw a circle:**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**circle(300,200,200);**

**getch();**

**return 0;**

**}**

**//Program to draw a rectangle**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**rectangle(150,250,450,350);**

**getch();**

**return 0;**

**}**

**//Program to draw a ellipse**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**ellipse(150,150,0,360,70,150);**

**getch();**

**return 0;**

**}**

**//Program to draw a triangle:**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**//Draw three line segments**

**line(150, 150, 450, 150);**

**line(150, 150, 300, 300);**

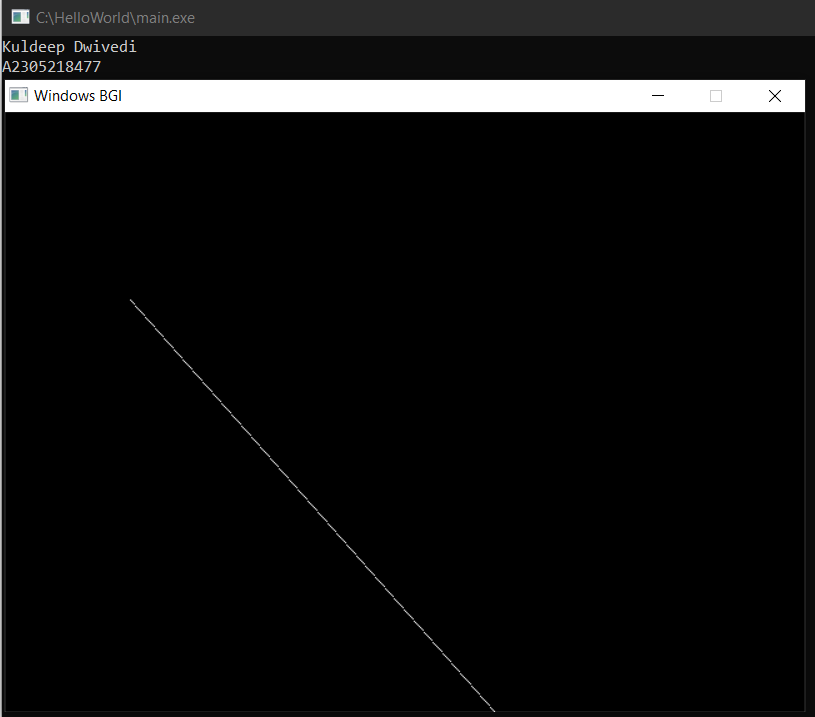
**line(450, 150, 300, 300);**

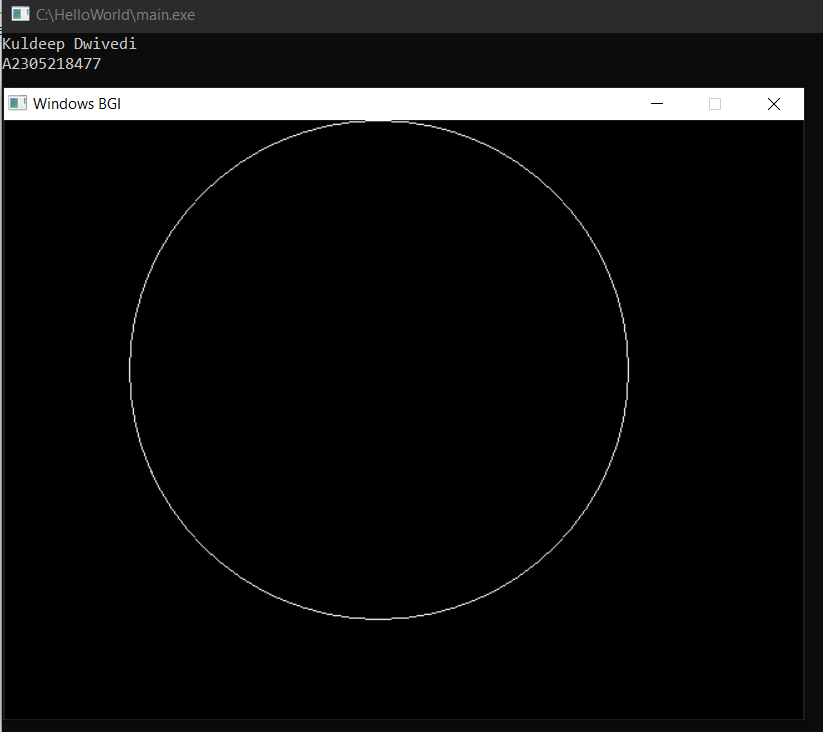
**getch();**

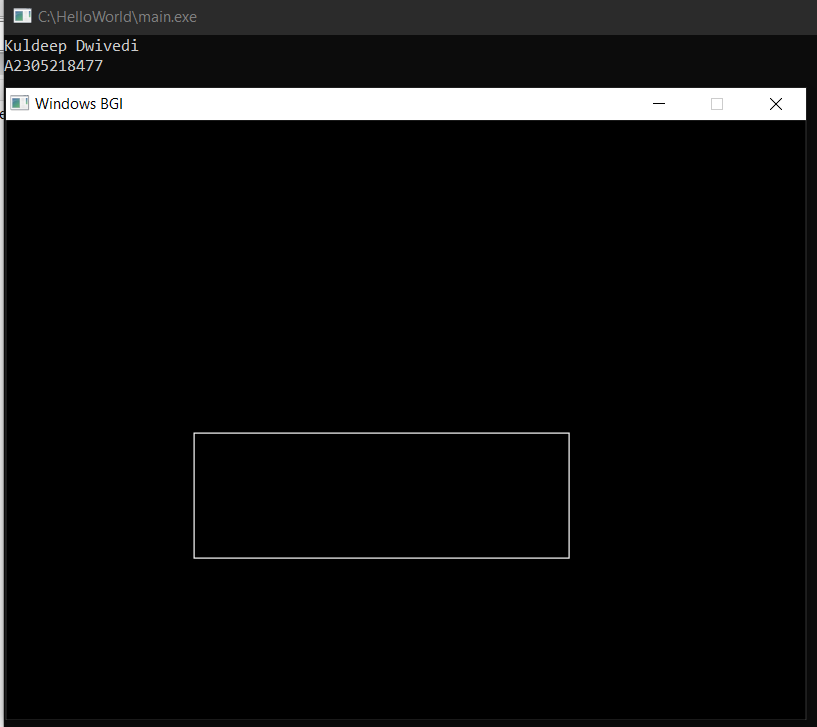
**return 0;**

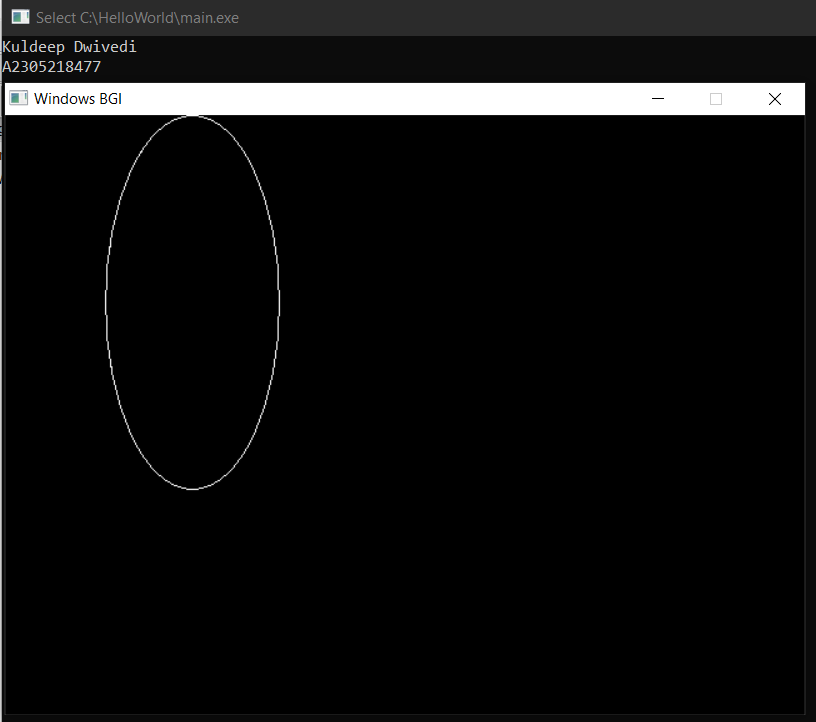
**}**

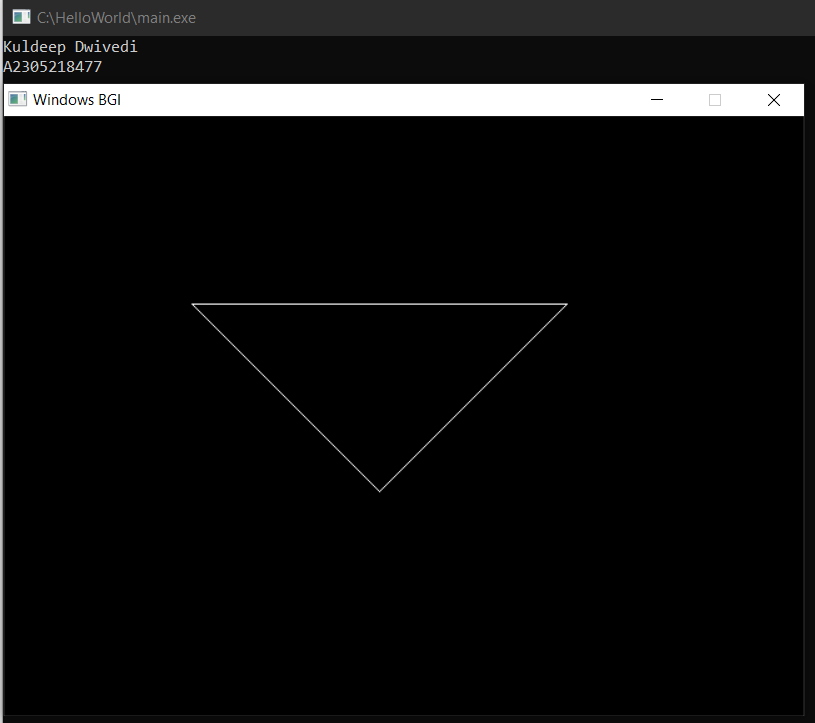
**Output:**

****

****

****

****

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 2**

**Date: 15/1/2021**

**Objective:** Write a Program to draw a line using DDA Algorithms

**Software Used:**Turbo C++

**Theory:**

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**int x1, y1, x2, y2;**

**cout<<"Enter x1 and y1"<<endl;**

**cin>>x1>>y1;**

**cout<<"Enter x2 and y2"<<endl;**

**cin>>x2>>y2;**

**float x, y;**

**x = float(x1);**

**y = float(y1);**

**float m;**

**m = (float)((y2-y1)/(x2-x1));**

**putpixel(x1,y1,WHITE);**

**while(x < x2)**

**{**

**x = x + 1;**

**y = y + m;**

**putpixel(round(x), round(y),WHITE);**

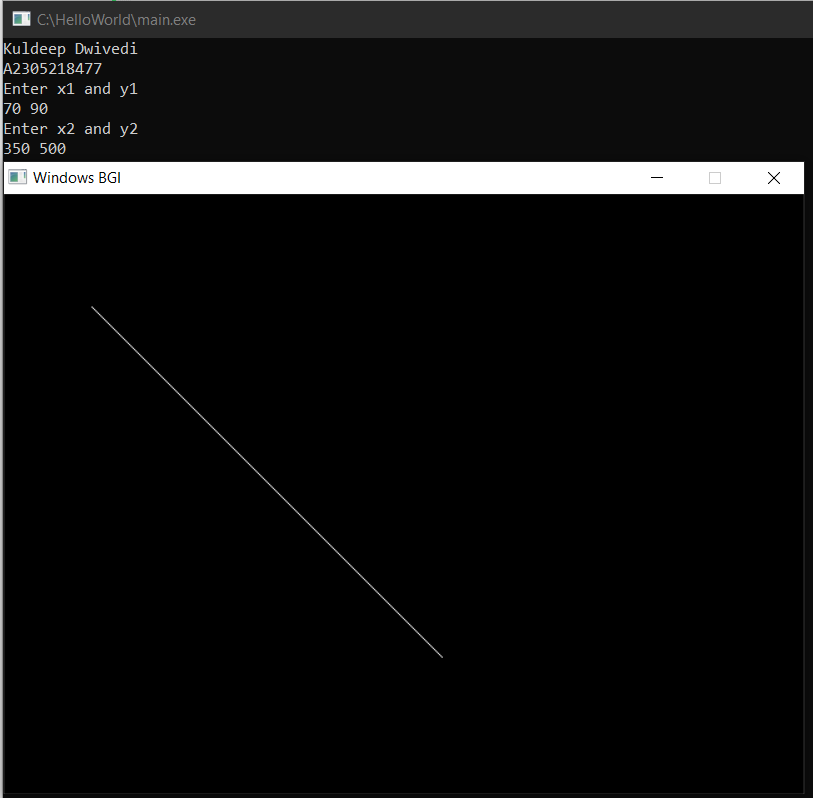
**}**

**getch();**

**return 0;**

**}**

**Output:**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

**EXPERIMENT 3**

**Date: 22/1/2021**

**Objective:** Write a Program to draw a line using Bresenhem's Algorithm

**Software Used:** Turbo C++

**Theory:**

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**int x1,y1,x2,y2;**

**cout<<"Enter x1 and y1"<<endl;**

**cin>>x1>>y1;**

**cout<<"Enter x2 and y2"<<endl;**

**cin>>x2>>y2;**

**int d;**

**int exch = 0;**

**int x\_diff = (x2-x1);**

**int y\_diff = (y2-y1);**

**int s1 = (x2-x1)/abs(x2-x1);**

**int s2 = (y2-y1)/abs(y2-y1);**

**if(y\_diff > x\_diff)**

**{**

**int temp = x\_diff;**

**x\_diff = y\_diff;**

**y\_diff = temp;**

**exch = 1;**

**}**

**else**

**exch = 0;**

**d = 2\*y\_diff - x\_diff;**

**int x = x1;**

**int y = y1;**

**for(int i = 1; i < x\_diff; i++)**

**{**

**putpixel(x,y,WHITE);**

**while(d >= 0)**

**{**

**if(exch == 1)**

**x = x + s1;**

**else**

**y = y + s2;**

**d = d - (2\*x\_diff);**

**}**

**if(exch == 1)**

**y = y + s2;**

**else**

**x = x + s1;**

**d = d + (2\*y\_diff);**

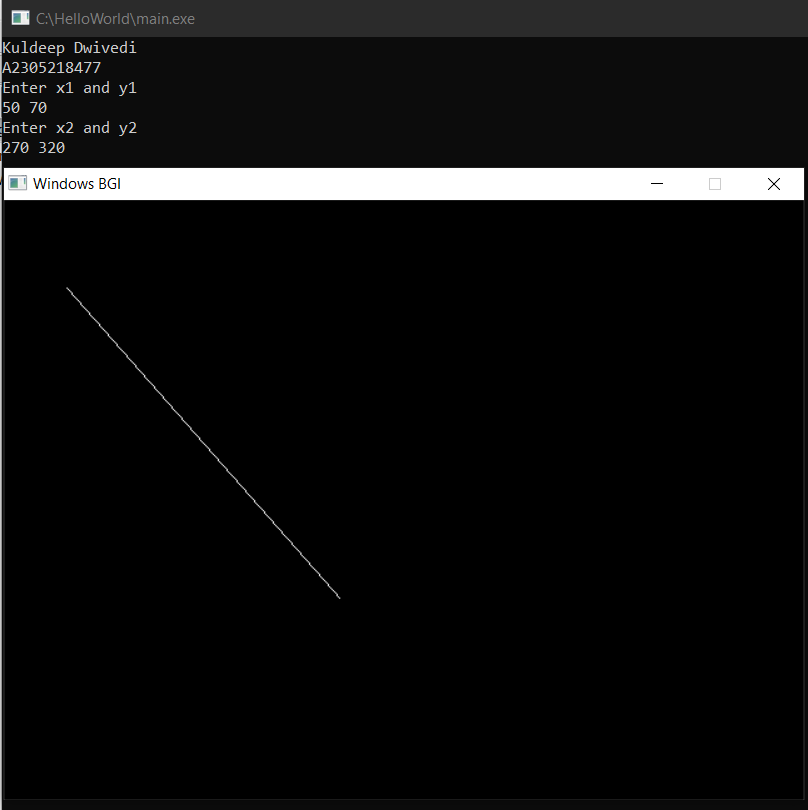
**}**

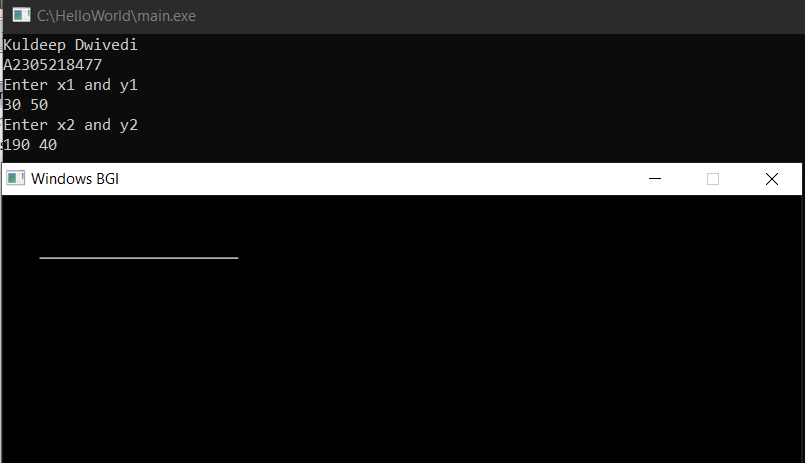
**getch();**

**return 0;**

**}**

**Output:**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 4**

**Date: 29/1/2021**

**Objective:** Write a Program to draw a circle using Bresenhem's Algorithm

**Software Used:**Turbo C++

**Theory:**

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**using namespace std;**

**void put\_circle\_pixel(int x, int y, int h, int k)**

**{**

**putpixel(h + x, k + y, WHITE);**

**putpixel(h - x, k + y, WHITE);**

**putpixel(h - x, k - y, RED);**

**putpixel(h + x, k - y, RED);**

**putpixel(h + y, k + x, RED);**

**putpixel(h - y, k + x, RED);**

**putpixel(h - y, k - x, RED);**

**putpixel(h + y, k - x, RED);**

**}**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**int r, p,x,y,h,k;**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A230521847"<<endl;**

**cout<<"Mid-Point Circle Algorithm"<<endl;**

**cout<<"Enter the radius"<<endl;**

**cin>>r;**

**cout<<"Enter the co-ordinates of center"<<endl;**

**cin>>h>>k;**

**p = 1-r;**

**x = 0;**

**y = r;**

**do**

**{**

**put\_circle\_pixel(x,y, h, k);**

**if(p < 0)**

**{**

**x = x + 1;**

**y = y;**

**p = p+ 2\*x +1;**

**}**

**else**

**{**

**x = x + 1;**

**y = y - 1;**

**p = p + 2\*x - 2\*y + 1;**

**}**

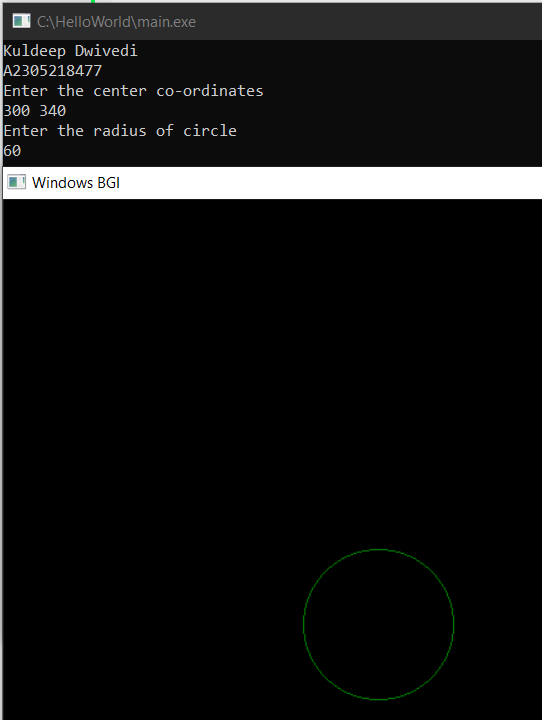
**}while(x < y);**

**getch();**

**return 0;**

**}**

**Output:**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 5**

**Date:5/2/2020**

**Objective:** Write a program to draw a circle using mid-point algorithm

**Software Used:**Turbo C++

**Theory:**

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**using namespace std;**

**void put\_circle\_pixel(int x, int y, int h, int k)**

**{**

**putpixel(h + x, k + y, WHITE);**

**putpixel(h - x, k + y, WHITE);**

**putpixel(h - x, k - y, RED);**

**putpixel(h + x, k - y, RED);**

**putpixel(h + y, k + x, RED);**

**putpixel(h - y, k + x, RED);**

**putpixel(h - y, k - x, RED);**

**putpixel(h + y, k - x, RED);**

**}**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**int r, p,x,y,h,k;**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**cout<<"Mid-Point Circle Algorithm"<<endl;**

**cout<<"Enter the radius"<<endl;**

**cin>>r;**

**cout<<"Enter the co-ordinates of center"<<endl;**

**cin>>h>>k;**

**p = 1-r;**

**x = 0;**

**y = r;**

**do**

**{**

**put\_circle\_pixel(x,y, h, k);**

**if(p < 0)**

**{**

**x = x + 1;**

**y = y;**

**p = p+ 2\*x +1;**

**}**

**else**

**{**

**x = x + 1;**

**y = y - 1;**

**p = p + 2\*x - 2\*y + 1;**

**}**

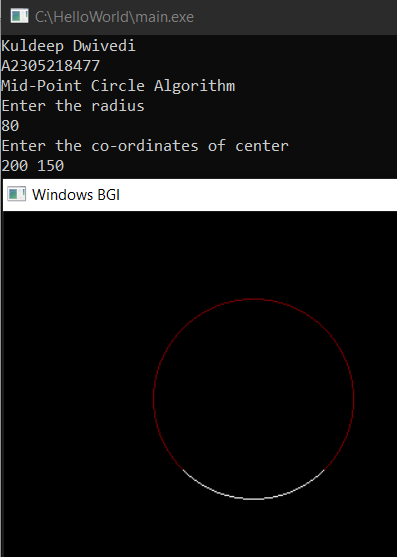
**}while(x < y);**

**getch();**

**return 0;**

**}**

**Output:**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 6**

**Date:12/2/2020**

**Objective:** Write a program to draw a ellipse using mid-point algorithm

**Software Used:**Turbo C++

**Theory:**

Mid-point Ellipse algorithm is used to draw an ellipse in computer graphics.

Midpoint ellipse algorithm plots(finds) points of an ellipse on the first quadrant by dividing the quadrant into two regions. Each point(x, y) is then projected into other three quadrants (-x, y), (x, -y), (-x, -y) i.e. it uses 4-way symmetry.

**Decision Parameter:**

Initially, we have two decision parameters p10 in region 1 and p20 in region 2.  
These parameters are defined as : p10 in region 1 is given as :

p10=ry2+1/4rx2-rx2ry

**Code:**

**#include<iostream>**

**#include<dos.h>**

**#include<conio.h>**

**#include<math.h>**

**#include<graphics.h>**

**using namespace std;**

**void display (int xs1, int ys1, int x, int y);**

**void ellips1(int xs1,int ys1,int rx, int ry)**

**{**

**int x,y;**

**float d1,d2,dx,dy;**

**x = 0; // take start position as (0,ry)**

**y = ry; // finding decision parameter d1**

**d1 = pow(ry,2) - (pow(rx,2) \* ry) + (0.25 \* pow(rx,2));**

**dx = 2 \* pow(ry,2) \* x;**

**dy = 2 \* pow(rx,2) \* y;**

**do // region one**

**{**

**display(xs1,ys1,x,y);**

**if(d1<0)**

**{**

**x++;**

**dx = dx + (2 \* (pow(ry,2)));**

**d1 = d1 + dx +(pow(ry,2));**

**}**

**else**

**{**

**x++;**

**y--;**

**dx = dx + (2 \* (pow(ry,2)));**

**dy = dy - (2 \* (pow(rx,2)));**

**d1 = d1 + dx - dy + (pow(ry,2));**

**}**

**}while(dx<dy); // change over condition for region-2**

**d2 = (pow(ry,2)\*pow((x+0.5),2)) + (pow(rx,2)\*pow((y-1),2)) - (pow(rx,2)\*pow(ry,2));**

**do**

**{**

**display(xs1,ys1,x,y);**

**if(d2>0)**

**{**

**x = x;**

**y--;**

**dy = dy - (2 \* (pow(rx,2)));**

**d2 = d2 - dy + pow(rx,2);**

**}**

**else**

**{**

**x++;**

**y--;**

**dy = dy - (2 \* (pow(rx,2)));**

**dx = dx + (2 \* (pow(ry,2)));**

**d2 = d2 +dx - dy + pow(rx,2);**

**}**

**}while(y>0);**

**}**

**void display(int xs,int ys,int x,int y)**

**{**

**putpixel(xs+x,ys+y,WHITE); // plot points by using 4 point symmetry**

**putpixel(xs-x,ys-y,WHITE);**

**putpixel(xs+x,ys-y,WHITE);**

**putpixel(xs-x,ys+y,WHITE);**

**}**

**int main(void)**

**{**

**int xs1,ys1;**

**float rx1,ry1;**

**int gd = DETECT,gm; // Initialise the graphics system**

**initgraph(&gd,&gm,(char\*)"");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**cout<<"Mid-Point Ellipse"<<endl;**

**cout<<"Enter the Center Co-ordinates\n";**

**cin>>xs1>>ys1;**

**cout<<"Enter the minor and major radius";**

**cin>>rx1>>ry1;**

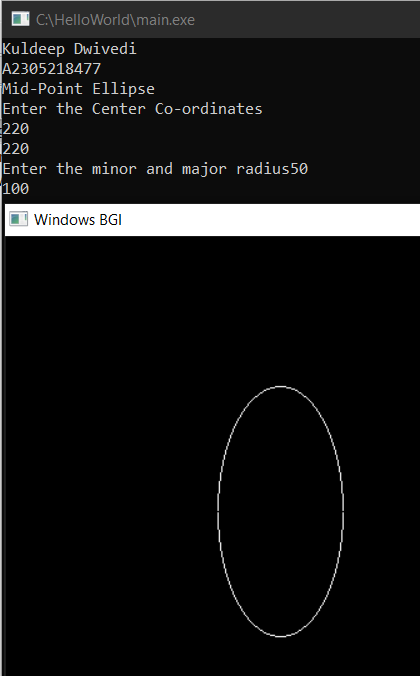
**ellips1(xs1,ys1,rx1,ry1);**

**getch();**

**closegraph();**

**}**

**Output:**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 7**

**Date: 19/2/2021**

**Objective:** Write a program for translation and scaling of the triangle

**Software Used:**Turbo C++

**Theory:**

In computer graphics, scaling is a process of modifying or altering the size of objects.

* Scaling may be used to increase or reduce the size of object.
* Scaling subjects the coordinate points of the original object to change.
* Scaling factor determines whether the object size is to be increased or reduced.
* If scaling factor > 1, then the object size is increased.
* If scaling factor < 1, then the object size is reduced.

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"\t\t2D SCALING"<<endl;**

**cout<<"\tKuldeep Dwivedi A2305218477"<<endl;**

**int tri[3][2];**

**float scaling[2][2] = {0};**

**float Sx, Sy;**

**cout<<"Enter the co-ordinates of triangle"<<endl;**

**for(int i = 0; i < 3; i++)**

**{**

**cout<<"Co-ordinate "<<(i+1)<<"\t";**

**for(int j = 0; j < 2; j++)**

**cin>>tri[i][j];**

**}**

**cout<<"Enter the value of Sx\t";**

**cin>>Sx;**

**cout<<"Enter the value of Sy\t";**

**cin>>Sy;**

**scaling[0][0] = Sx;**

**scaling[1][1] = Sy;**

**int new\_tri[3][2] = {0};**

**for(int i = 0; i < 3; i++)**

**{**

**for(int j = 0; j < 2; j++)**

**{**

**for(int k = 0; k < 2; k++)**

**{**

**new\_tri[i][j] += tri[i][k]\*scaling[k][j];**

**}**

**}**

**}**

**int arr[] = {tri[0][0],tri[0][1],tri[1][0],tri[1][1],tri[2][0],tri[2][1],tri[0][0],tri[0][1]};**

**drawpoly(4,arr);**

**int arr1[] = {new\_tri[0][0],new\_tri[0][1],new\_tri[1][0],**

**new\_tri[1][1],new\_tri[2][0],new\_tri[2][1],new\_tri[0][0],new\_tri[0][1]};**

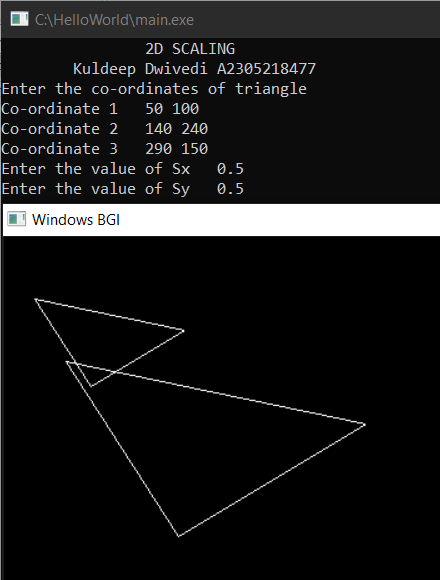
**drawpoly(4,arr1);**

**getch();**

**return 0;**

**}**

**Output:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 8**

**Date: 5/3/2021**

**Objective:** Write a program for rotation of the triangle

**Software Used:**Turbo C++

**Theory:**

In Computer graphics, 2D Translation is a process of moving an object from one position to another in a two dimensional plane. Consider a point object O has to be moved from one position to another in a 2D plane.

Let-

* Initial coordinates of the object O = (Xold, Yold)
* New coordinates of the object O after translation = (Xnew, Ynew)
* Translation vector or Shift vector = (Tx, Ty)

Given a Translation vector (Tx, Ty)-

* Tx defines the distance the Xold coordinate has to be moved.
* Ty defines the distance the Yold coordinate has to be moved.

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**cout<<"\t\t2D TRANSLATION"<<endl;**

**cout<<"\tKuldeep Dwivedi A2305218477"<<endl;**

**cout<<"Green- Before\n White - After"<<endl;**

**int tri[3][2];**

**float Tx, Ty;**

**cout<<"Enter the co-ordinates of triangle"<<endl;**

**for(int i = 0; i < 3; i++)**

**{**

**cout<<"Co-ordinate "<<(i+1)<<"\t";**

**for(int j = 0; j < 2; j++)**

**cin>>tri[i][j];**

**}**

**cout<<"Enter the value of Tx\t";**

**cin>>Tx;**

**cout<<"Enter the value of Ty\t";**

**cin>>Ty;**

**int arr[] = {tri[0][0],tri[0][1],tri[1][0],tri[1][1],tri[2][0],tri[2][1],tri[0][0],tri[0][1]};**

**drawpoly(4,arr);**

**int arr1[] = {tri[0][0]+Tx,tri[0][1] + Ty,tri[1][0]+Tx,tri[1][1]+Ty,tri[2][0]+Tx,tri[2][1]+Ty,tri[0][0]+Tx,tri[0][1]+Ty};**

**setcolor(WHITE);**

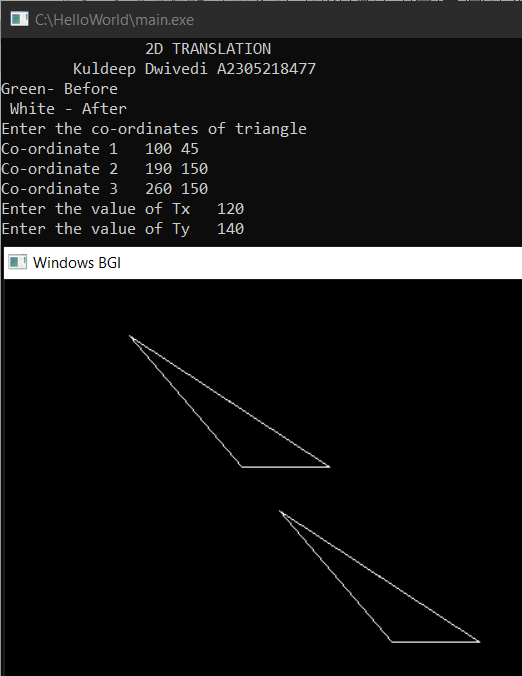
**drawpoly(4,arr1);**

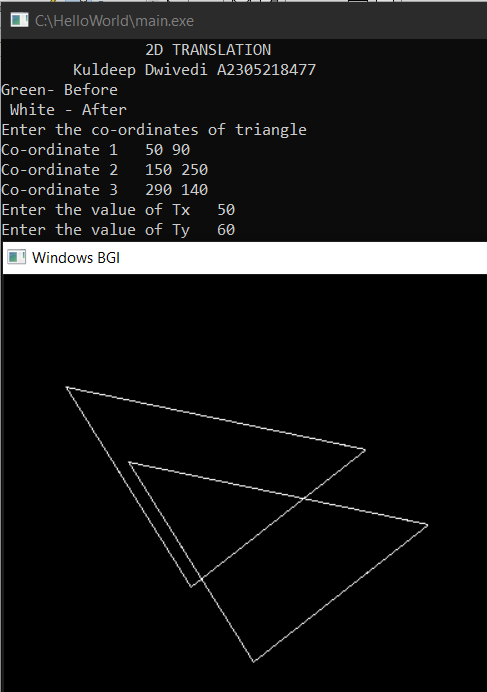
**getch();**

**return 0;**

**}**

**Output:**

****

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 9**

**Date: 12/3/2021**

**Objective:** Write a program for reflection of the triangle

**Software Used:**Turbo C++

**Theory:**

In Computer graphics, 2D Rotation is a process of rotating an object with respect to an angle in a two dimensional plane.

Consider a point object O has to be rotated from one angle to another in a 2D plane.

Let-

* Initial coordinates of the object O = (Xold, Yold)
* Initial angle of the object O with respect to origin = Φ
* Rotation angle = θ
* New coordinates of the object O after rotation = (Xnew, Ynew)

This rotation is achieved by using the following rotation equations-

* Xnew = Xold x cosθ – Yold x sinθ
* Ynew = Xold x sinθ + Yold x cosθ

**Code:**

**#include <iostream>**

**#include<graphics.h>**

**#include<math.h>**

**#define PI 3.1415926**

**using namespace std;**

**int main()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"");**

**int x1,y1,x2,y2,x3,y3;**

**cout<<"\t\tKuldeep A2305218477"<<endl;**

**cout<<"\t\t2-D Rotation"<<endl;**

**cout<<"Before Rotation- White\n After Rotation- Green"<<endl;**

**int angle;**

**double cosine,sine;**

**cout<<"Enter the co-ordinates of triangle"<<endl;**

**cout<<"Co-ordinate 1:\t";**

**cin>>x1>>y1;**

**cout<<"Co-ordinate 2:\t";**

**cin>>x2>>y2;**

**cout<<"Co-ordinate 3:\t";**

**cin>>x3>>y3;**

**line(x1,y1,x2,y2);**

**line(x2,y2,x3,y3);**

**line(x3,y3,x1,y1);**

**cout<<"Enter the angle of rotation"<<endl;**

**cin>>angle;**

**cosine = cos(angle\*PI/180);**

**sine = sin(angle\*PI/180);**

**x1 = floor(x1 \* cosine + y1 \* sine);**

**y1 = floor(-x1 \* sine + y1 \* cosine);**

**x2 = floor(x2 \* cosine + y2 \* sine);**

**y2 = floor(-x2 \* sine + y2 \* cosine);**

**x3 = floor(x3 \* cosine + y3 \* sine);**

**y3 = floor(-x3 \* sine + y3 \* cosine);**

**setcolor(GREEN);**

**line(x1,y1,x2,y2);**

**line(x2,y2,x3,y3);**

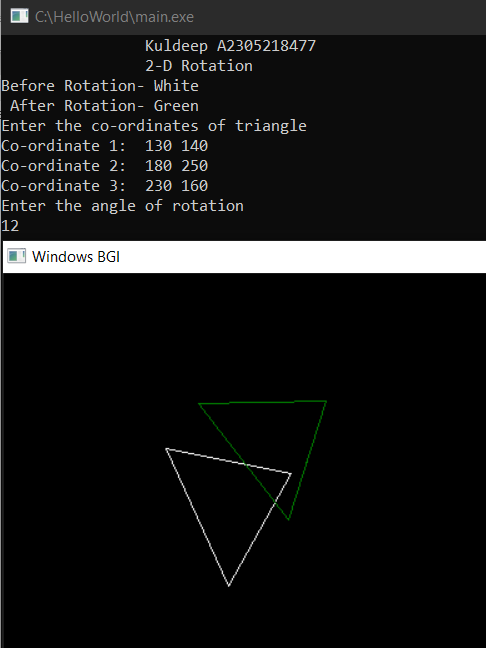
**line(x3,y3,x1,y1);**

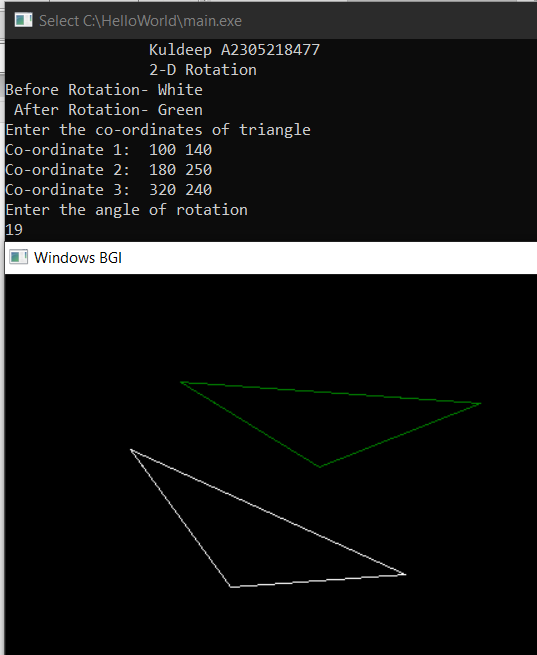
**getch();**

**return 0;**

**}**

**Output:**

****

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

# **EXPERIMENT 10**

**Date: 19/3/2021**

**Objective:** Write a program for Cohen- Sutherland clipping algorithm

**Software Used:**Turbo C++

**Theory:**

The primary use of clipping in computer graphics is to remove objects, lines, or line segments that are outside the viewing pane. The viewing transformation is insensitive to the position of points relative to the viewing volume − especially those points behind the viewer − and it is necessary to remove these points before generating the view.

**Cohen-Sutherland Line Clipping Algorithm:** This algorithm uses the clipping window. The minimum coordinate for the clipping region is (*XWmin*,*YWmin*) and the maximum coordinate for the clipping region is (*XWmax*,*YWmax*). We will use 4-bits to divide the entire region. These 4 bits represent the Top, Bottom, Right, and Left of the region as shown in the following figure. Here, the **TOP** and **LEFT** bit is set to 1 because it is the **TOP-LEFT** corner.

**Code:**

**#include<iostream>**

**#include<conio.h>**

**#include<graphics.h>**

**#include<math.h>**

**using namespace std;**

**void Window()**

**{**

**line (200,200,350,200);**

**line(350,200,350,350);**

**line(200,200,200,350);**

**line(200,350,350,350);**

**}**

**void Code(char c[4],float x,float y)**

**{ c[0]=(x<200)?'1':'0';**

**c[1]=(x>350)?'1':'0';**

**c[2]=(y<200)?'1':'0';**

**c[3]=(y>350)?'1':'0';**

**}**

**void Clipping (char c[],char d[],float &x,float &y,float m)**

**{**

**int flag=1,i=0;**

**for (i=0;i<4;i++)**

**{**

**if(c[i]!='0' && d[i]!='0')**

**{**

**flag=0;**

**break;**

**}**

**if(flag)**

**{**

**if(c[0]!='0')**

**{**

**y=m\*(200-x)+y;**

**x=200;**

**}**

**else if(c[1]!='0')**

**{**

**y=m\*(350-x)+y;**

**x=350;**

**}**

**else if(c[2]!='0')**

**{**

**x=((200-y)/m)+x;**

**y=200;**

**}**

**else if(c[3]!='0')**

**{**

**x=((350-y)/m)+x;**

**y=350;**

**}**

**}**

**if (flag==0)**

**cout<<"Line lying outside";**

**}**

**}**

**int main()**

**{**

**int gdriver = DETECT, gmode, errorcode;**

**float x1,y1,x2,y2;**

**float m;**

**char c[4],d[4];**

**initgraph(&gdriver, &gmode, "//Turboc3//bgi");**

**cout<<"Kuldeep Dwivedi"<<endl;**

**cout<<"A2305218477"<<endl;**

**cout<<"Enter coordinates";**

**cin>>x1>>y1>>x2>>y2;**

**cout<<"Before clipping";**

**Window();**

**line(x1,y1,x2,y2);**

**getch();**

**cleardevice();**

**m=float((y2-y1)/(x2-x1));**

**Code(c,x1,y1);**

**Code(d,x2,y2) ;**

**Clipping(c,d,x1,y1,m);**

**Clipping(d,c,x2,y2,m);**

**cout<<"After Clipping";**

**Window();**

**line(x1,y1,x2,y2);**

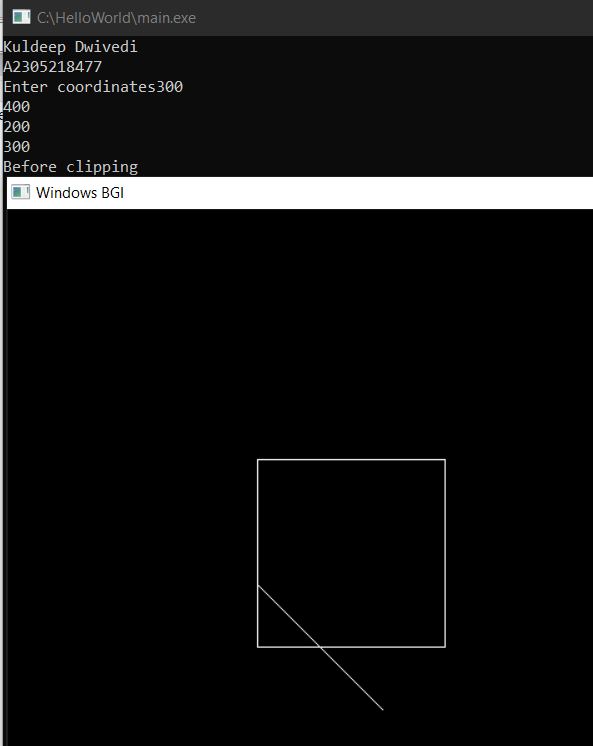
**getch();**

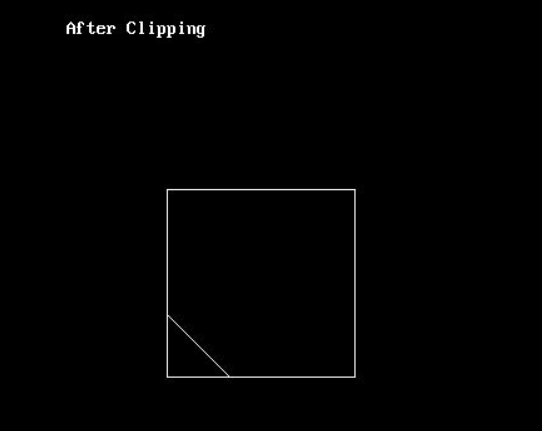
**closegraph();**

**return 0;**

**}**

**Output:**

****

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP) | | | | |
| Programme | B.Tech CSE | | Course Name | Computer Graphics |
| Course Code | [CSE203] | | Semester | 6 |
| Student Name | TTTTTT | | Enrollment No. | A23009100000 |
| Marking Criteria | | | | |
| Criteria | Total Marks | Marks Obtained | | Comments |
| Concept (A) | 2 |  | |  |
| Implementation (B) | 2 |  | |  |
| Performance (C) | 2 |  | |  |
| Total | 6 |  | |  |

**Viva Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Internal Assessment (Viva Component) Sheet for Lab Experiment Department of Computer Science & Engineering Amity University, Noida (UP)** | | | |
| Programme | B. Tech CSE | Course Name | Computer Graphics |
| Course Code | [CSE203] | Semester | 6 |
| Student Name | TTTTT | Enrollment No. | A23000000 |
| **Marking Criteria** | | | |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Clarity of the Subject (H) | 4 |  |  |
| Quality of theoretical Discussion (I) | 6 |  |  |
| Total | 10 |  |  |