I

Name _____

Roll No. _____ Year 20____20____

Exam Seat No. _____

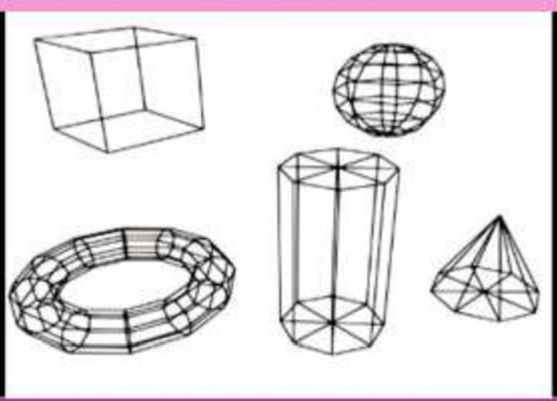
COMPUTER GROUP | SEMESTER - III | DIPLOMA IN ENGINEERING AND TECHNOLOGY

A LABORATORY MANUAL

FOR

COMPUTER GRAPHICS

(22318)



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI
(Autonomous) (ISO 9001 : 2015) (ISO / IEC 27001 : 2013)

VISION

To ensure that the Diploma level Technical Education constantly matches the latest requirements of technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

MISSION

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the changing technological and environmental challenges.

QUALITY POLICY

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

CORE VALUES

MSBTE believes in the followings:

- Education industry produces live products.
- Market requirements do not wait for curriculum changes.
- Question paper is the reflector of academic standards of educational organization.
- Well designed curriculum needs effective implementation too.
- Competency based curriculum is the backbone of need based program.
- Technical skills do need support of life skills.
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.

Computer Graphics

(22318)

Semester-III

(CO/CM/CW)

Maharashtra State
Board of Technical Education, Mumbai
(Autonomous) (ISO:9001:2015) (ISO/IEC 27001:2013)

Maharashtra State Board of Technical Education,
(Autonomous) (ISO : 9001 : 2015) (ISO/IEC 27001 : 2013)
4th Floor, Government Polytechnic Building, 49, Kherwadi,
Bandra (East), Mumbai - 400051.
(Printed on June, 2018)

BOARD OF TECHNICAL EDUCATION

Certificate

This is to certify that Mr. / Ms. ……..…… Roll No. ……..…., of Third Semester of Diploma in ……..…… of Institute, ……..…….……..… (Code: ……...) has completed the term work satisfactorily in course **Computer Graphics (22318)** for the academic year 20…….. to 20…….. as prescribed in the curriculum.

Place: …… Enrollment No:.……

Date: …… Exam. Seat No: ……

Subject Teacher **Head of the Department** **Principal**

Seal of Institution

Preface

The primary focus of any engineering laboratory/ field work in the technical education system is to develop the much needed industry relevant competencies and skills. With this in view, MSBTE embarked on this innovative 'I' Scheme curricula for engineering diploma programmes with outcome-base education as the focus and accordingly, relatively large amount of time is allotted for the practical work. This displays the great importance of laboratory work making each teacher; instructor and student to realize that every minute of the laboratory time need to be effectively utilized to develop these outcomes, rather than doing other mundane activities. Therefore, for the successful implementation of this outcome-based curriculum, every practical has been designed to serve as a ***'vehicle'*** to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'I' scheme laboratory manual development team designed the practical's to ***focus*** on the ***outcomes***, rather than the traditional age old practice of conducting practical's to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read through the concerned practical procedure that they will do the next day and understand the minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. The students will then become aware about the skills they will achieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

This course provides an introduction to the principles of Computer graphics. In particular, the course will consider methods for object design, transformation, scan conversion, visualization and modeling of real world. The emphasis of the course will be placed on understanding how the various elements that under-lie Computer graphics (algebra, geometry, algorithms) interact in the design of graphics software systems and also enables student to create impressive graphics easily and efficiently.

Although best possible care has been taken to check for errors (if any) in this laboratory manual, perfection may elude us as this is the first edition of this manual. Any errors and suggestions for improvement are solicited and highly welcome.

Programme Outcomes (POs) to be achieved through Practical of this Course

PO1. **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer engineering problem.

PO2. **Discipline knowledge:** Apply Computer Engineering knowledge to solve broad- based Computer Engineering related problems.

PO3. **Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Computer Engineering related problems.

PO4. **Engineering tools:** Apply appropriate Computer Engineering related techniques/ tools with an understanding of the limitations.

PO7. **Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of the practice also in the field of Computer Engineering.

PO8. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

PO9. **Communication:** Communicate effectively in oral and written form.

PO10. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes in the Computer Engineering field and allied industry.

****

List of Industry Relevant Skills

- The following industry relevant skills of the competency **' Develop programs using core graphical concepts'** are expected to be developed in you by undertaking the practical's of this laboratory manual.

1. Draw various graphics objects.
2. Design CAD-CAM software.
3. Design games.
4. Design animations.
5. Perform 2D and 3D transformations.

Practical- Course Outcome matrix

Course Outcomes (COs)

a. Manipulate visual and geometric information of images.
b. Implement standard algorithms to draw various graphics objects using C program.
c. Develop programs for 2-D and 3-D Transformations.
d. Use projections to visualize objects on view plane.
e. Implement various clipping algorithms.
f. Develop programs to create curves using algorithms.

S. No.	Practical Outcome	CO a.	CO b.	CO c.	CO d.	CO e.	CO f.
1	Write Programs to draw following graphics object using built-in "C" functions. a) Pixel b) Lines c) Circles d) Rectangle e) Ellipse	√	-	-	-	-	-
2	Implement DDA algorithm to draw line.	-	√	-	-	-	-
3	Implement Bresennham's algorithm to draw line	-	√	-	-	-	-
4	Implement Bresennham's algorithm to draw a circle.	-	√	-	-	-	-
5	Write a program to fill Polygon using Flood fill method.	-	√	-	-	-	-
6	Write a program to fill Polygon using Boundary fill method.	-	√	-	-	-	-
7	Write a program in C to perform two-dimensional transformation (Translation and Scaling)	-	√	√	-	-	-
8	Write a program in C to perform two dimensional transformation (Rotation)	-	√	√	-	-	-
9	Write a program in C to perform two-dimensional transformation (Reflection and Shearing)	-	√	√	-	-	-
10	Write a program in C to perform three-dimensional transformation for a) Translation b) Scaling	-	√	√	-	-	-

11	Write a program in C to perform three-dimensional transformation for a) Rotation	-	√	√	-	-	-
12	Write a program to clip line using Cohen-Sutherland algorithm	-	√	-	-	√	-
13	Write a program to clip line using Midpoint subdivision algorithm	-	√	-	-	√	-
14	Write a program to clip polygon using Sutherland –Hodgeman algorithm.	-	√	-	-	√	-
15	Write a program to draw Hilbert's Curve	-	√	-	-	-	√
16	Write a program to draw Koch curve and Bezier curves	-	√	-	-	-	√

Guidelines to Teachers

1. There will be two sheets of blank pages after every practical for the student to report other matters (if any), which is not mentioned in the printed practical.
2. For difficult practical if required, teacher could provide the demonstration of the practical emphasizing of the skills which the student should achieve.
3. Teachers should give opportunity to students for hands-on after the demonstration.
4. Assess the skill achievement of the students and COs of each unit.
5. One or two questions ought to be added in each practical for different batches. For this teachers can maintain various practical related question banks for each course.
6. For effective implementation and attainment of practical outcomes, teacher ought to ensure that in the beginning itself of each practical, students must read through the complete write-up of that practical sheet.
7. During practical, ensure that each student gets chance and takes active part in taking observations/ readings and performing practical.
8. Teacher ought to assess the performance of students continuously according to the MSBTE guidelines.

Instructions for Students

Note: Kindly do add specific instructions for students for effective implementation of upon your course, if practical depending needed.

1. For incidental writing on the day of each practical session every student should maintain a ***dated log book*** for the whole semester, apart from this laboratory manual which s/he has to ***submit for assessment to the teacher*** in the next practical session.
2. For effective implementation and attainment of practical outcomes, in the beginning it of each practical, students need to read through the complete write-up including the practical related questions and assessment scheme of that practical sheet.
3. Student ought to refer the reference books, lab manuals, etc.
4. Student should not hesitate to ask any difficulties they face during the conduct of practical.

Content Page

List of Practical's and Progressive Assessment Sheet

Sr. No	Practical Outcome	Page No.	Date of perfor mance	Date of submit-ssion	Assess-ment marks(25)	Dated sign. of teacher	Remarks (if any)
1.	Write Programs to draw following graphics object using built-in "C" functions. a) Pixel b) Lines c) Circles d) Rectangle e) Ellipse	1					
2.	Implement DDA algorithm to draw line.	10					
3.	Implement Bresennham's algorithm to draw line	18					
4.	Implement Bresennham's algorithm to draw a circle.	25					
5.	Write a program to fill Polygon using Flood fill method.	33					
6.	Write a program to fill Polygon using Boundary fill method.	40					
7.	Write a program in C to perform two-dimensional transformation (Translation and Scaling)	48					
8.	Write a program in C to perform two dimensional transformation (Rotation)	56					
9.	Write a program in C to perform two-dimensional transformation (Reflection and Shearing)	65					
10.	Write a program in C to perform three-dimensional transformation for a) Translation b) Scaling	74					
11.	Write a program in C to perform three-dimensional transformation for a) Rotation	82					
12.	Write a program to clip line using Cohen-Sutherland algorithm	90					

Sr. No	Practical Outcome	Page No.	Date of perfor mance	Date of submit-ssion	Assess-ment marks(25)	Dated sign. of teacher	Remarks (if any)
13.	Write a program to clip line using Midpoint subdivision algorithm	100					
14.	Write a program to clip polygon using Sutherland –Hodgeman algorithm.	107					
15.	Write a program to draw Hilbert's Curve	116					
16.	Write a program to draw Koch curve and Bezier curves	124					
Total							

- **To be transferred to Proforma of CIAAN-2017.**

Practical No. 1: Programs to draw basic graphic objects

I. Practical Significance

Computer graphics provides different graphics functions to draw various graphics objects. By using these basic graphic functions student will able to construct pixel, line, circle, rectangle, and ellipse and learn the graphics coordinate system to plot objects.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

In this practical, it is expected, to develop the following skills in students

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Manipulate visual and geometric properties of images.

V. Practical Outcome (POs):

- Write Programs to draw following graphics object using built-in "C" functions.

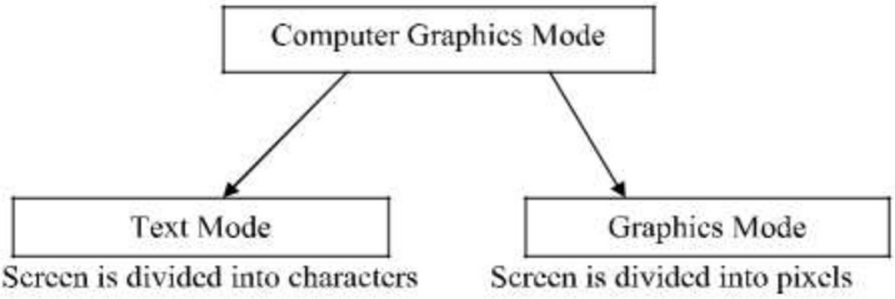
a) Pixel
b) Lines
c) Circles
d) Rectangle
e) Ellipse

VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Output Primitives: The **Primitives** are the simple geometric functions that are used to generate various **Computer Graphics** objects required by the User. Some most basic **Output primitives** are point-position (pixel), a straight line, circle, rectangle and ellipse.



Text Mode: Screen is divided into character positions. Screen is mapped as number of rows and columns.

Text mode functions:
putch( ): Displays single character at cursor position.
clrscr( ): Clears entire screen.
gotoxy ( ): Positions cursor to specifies location on screen.
Textcolor ( ): Apply color to text.

Graphics Mode: Screen is divided into pixels. While working in C programming language, default output mode is text mode. To draw graphics objects on screen display mode must be changed from text mode to graphics mode. To change from text mode to graphics mode, use functions below.
void initgraph(int *graphdriver, int graph mode, char *pathtodriver);

Some of the basic graphic mode functions are
#include<graphics.h> : The header file must be included for every graphics program

1. void initgraph(int *graphdriver, int graph mode, char *pathtodriver);
initgraph() function is used To change from text mode to graphics mode. The graphics driver and graphics mode are the parameters to this function. detectgraph() function is used to find out the graphics driver and graphics mode.

2. void closegraph();
closegraph() function closes the graphics mode, deallocates all memory allocated by graphics system and restores the screen to text mode.

3. putpixel(int x, int y, color);
putpixel() function is used to draw pixel at given position(x,y) and with given color.

4. line (int x1,int y1,int x2,int y2);
line function is used to draw a line from point(x1,y1)-Starting point to point (x2,y2)-Ending point

5. circle (int x, int y, int radius);
Circle function is used to draw a circle with center (x, y) and radius of a circle.

6. rectangle(int left, int top, int right, int bottom);
   Rectangle function is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle.

7. ellipse(int x, int y, start angle, end angle, int xradius, int yradius);
   Ellipse function is used to draw ellipse with center(x, y) ,starting angle, end angle, radius of x and y axis.

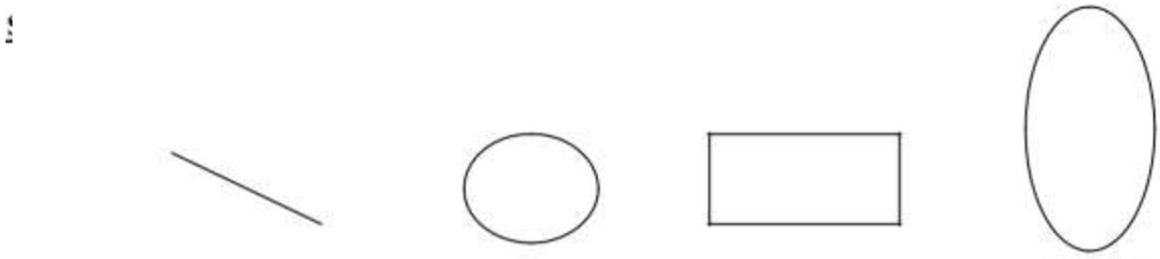
8. outtextxy(int x, int y, "text");
   outtextxy function is used to insert text at the given position(x,y).

Procedure:

1. List different basic graphics functions.
2. Select the proper function to draw respective object.
3. Use proper syntax for selected function.
4. Check the required format of output.

C Program Code:

```
#include<stdio.h>
#include<graphics.h>//must be included for every graphics program
#include<conio.h>
#include<dos.h> //for including delay function.
void main()
{
int gd=DETECT,gm; //gd=detects best available graphics driver, gm =graphics mode.
initgraph(&gd,&gm,"C:\\TurboC3\\BGI");// for initializing graph mode
// above 2 steps are must for every graphics program.
//declaration of any variables must be done before calling initgraph() function.
// next write code for producing requiring design or drawing object
putpixel(50,50,RED);//plot pixel of red color.
outtextxy(55,55,"PIXEL");
line(100,70,200,100);//draws a line segment.
outtextxy(220,110,"LINE");
circle(230,50,50);//draws a Circle.
outtextxy(290,120,"CIRCLE");
rectangle(300,40,400,100);/draws a Rectangle.
outtextxy(420,120,"PIXEL");
ellipse(450,50,0,360,100,175);//draws an Ellipse.
outtextxy(450,200,"ELLIPSE");
getch();
}
```



VIII. Algorithm

IX. Flow Chart

X. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XI. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in C to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIII. Result (Output of the Program)

.....

.....

.....

XIV. Conclusion(s)

.....

.....

.....

XV. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Find the minimum and maximum coordinates of screen.
2. What is graphics driver and graphics mode?
3. What is path of graphics driver?
4. Find error of the following code:

```
{
putpixel(20,20);
outtextxy(25,25,"PIXEL");
}
```

5. List four applications of Computer graphics.
6. Define terms pixel, line and raster scan graphics.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

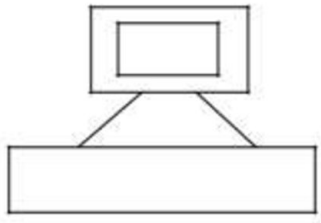
XVI. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

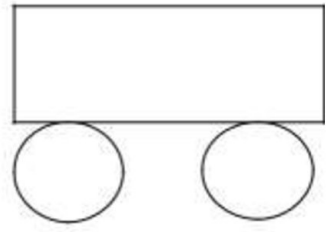
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a C program to display following objects

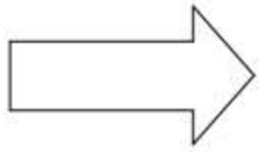
a.



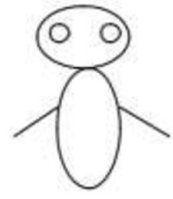
b.



c.

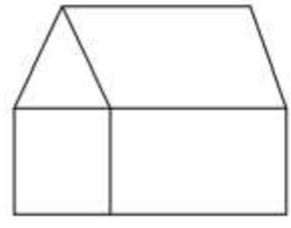


d.



e.

f.



(Space for Answers)

.....

.....

.....

XVII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2. http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. http://ecomputernotes.com/computer-graphics/basic-of-computer-graphics

XVIII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. .....
2. .....
3. .....
4. .....

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 2: Program to draw line using DDA algorithm.

I. Practical Significance

In Computer graphics, a digital differential analyzer (**DDA**) is used for interpolation of variables over an interval between start and end point. DDAs are used for rasterization of lines, triangles and polygons.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.

V. Practical Outcome (POs):

Implement DDA algorithm to draw line.

VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

DDA algorithm:

Digital Differential Analyzer (DDA) algorithm is the simple line generation algorithm. A line connects two end-points. It is a basic element in computer-graphics. To draw a line, you need two points between which you can draw a line. DDA is used for interpolation of variables over an interval between start and end point. DDAs are used for rasterization of lines, triangles and polygons. They can be extended to nonlinear functions, such as perspective correct texture mapping, quadratic curves, and traversing pixels.

In its simplest implementation for linear cases such as lines, the DDA algorithm interpolates values in interval by computing for each x_i the equations $x_i = x_{i-1} + 1$, $y_i = y_{i-1} + 1$.

Procedure:

Step 1: Read the input of the 2 end points of the line as (x1, y1) & (x2, y2) such that x1 != x2 and y1 != y2

Step 2: Calculate dx = x2 – x1 and dy = y2 – y1

Step 3:

```
if(dx>=dy)
        step=dx
else
        step=dy
```

Step 4: x_{in} = dx / step & y_{in} = dy / step

Step 5: x = x1 + 0.5 & y = y1 + 0.5

Step 6:

```
for(k = 0; k < step; k++)
{
        x = x + xin
        y = y + yin
        putpixel(x, y)
}
```

Sample Output of the source example:

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Define the term Rasterization.
2. Write slope intercept form of a line.
3. Write advantages and disadvantages of DDA algorithm.

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Give following values for every iteration of DDA algorithm to draw a line from(3,4) to(6,8)

dx	dy	step	xin	yin

ii. Give following values for every iteration of DDA algorithm to draw a line from (-5,-5) to(-12,-12)

	dy	step	xin	yin

(Space for Answers)

.....

.....

.....

.....

XVIII. References / Suggestions for further Reading

1. http://people.csail.mit.edu/fredo/Depiction/1_Introduction/reviewGraphics.pdf.
2. http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2.

XIX. Assessment Scheme

Performance indicators		Weightage
Process related(10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 3: Program to draw line using Bresenham's algorithm.

I. Practical Significance

Bresenham's line algorithm is an algorithm that determines the points of an *n*-dimensional raster that should be selected in order to form a close approximation to a straight line between two points. It is commonly used to draw line primitives in a bitmap image.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer engineering problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.

V. Practical Outcome (POs):

a) Implement Bresenham's algorithm to draw line.

VI. Relevant affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Bresenham's line algorithm

Bresenham's line algorithm determines the points of an *n*-dimensional raster that is selected to form a close approximation to a straight line between two points. It is used to draw line primitives in a bitmap image , as it uses only integer addition, subtraction and bit shifting. It is an incremental error algorithm. It is one of the earliest algorithms developed in the field of Computer graphics.

Consider a line with initial point (x1, y1) and terminal point (x2, y2) in device space. If $\Delta x = x2 - x1$ and $\Delta y = y2 - y1$, we define the driving axis (DA) to be the x-axis if $|\Delta x| \geq |\Delta y|$, and the y-axis if $|\Delta y| > |\Delta x|$. The DA is used as the "axis of control" for the algorithm and is the axis of maximum movement. Within the main loop of the algorithm, the coordinate corresponding to the DA is incremented by one unit. The coordinate corresponding to the other axis (usually denoted the passive axis or PA) is only incremented as needed.

Procedure

1. Input the two line end-points, storing the left end-point in (x_0,y_0)
2. Plot the point (x_0, y_0)
3. Calculate the constants Δx, Δy, $2\Delta y$, and $(2\Delta y - 2\Delta x)$ and get the first value for the decision parameter as: $P_0 = 2\Delta y - 2\Delta x$
4. At each x_k along the line, starting at $k=0$, perform the following test:
   If $p_k < 0$, the next point to plot is (x_k+1, y_k) and $p_{k+1} = p_k + 2\Delta y$
   Otherwise, the next point to plot is (x_k+1, y_k+1) and $p_{k+1} = p_k + 2\Delta y - 2\Delta x$
5. Repeat step 4 (Δx) times.

Sample Output of the source example:

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Explain the term decision parameter.
2. Write advantages of Bresenham's algorithm over DDA.

(Space for Answers)

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Give following values for every iteration of Bresenham's algorithm to draw a line from(3,4) to(6,8)

Δx	Δy	2Δx	2Δy	P

ii. Give following values for every iteration of Bresenham's algorithm to draw a line from(-6,-6) to(-14,-14)

Δx	Δy	2Δx	2Δy	P

(Space for Answers)

.....

.....

.....

.....

.....

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2. http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 4: Program to draw circle.

I. Practical Significance

It is not easy to display a continuous smooth arc on the Computer screen as our Computer screen is made of pixels organized in matrix form. So, to draw a circle on a Computer screen we should always choose the nearest pixels from a printed pixel so as they could form an arc. There are two popular algorithms for generating a circle – Bresenham's Algorithm and Midpoint Circle Algorithm. These algorithms are based on the idea of determining the subsequent points required to draw the circle.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.

V. Practical Outcome (POs):

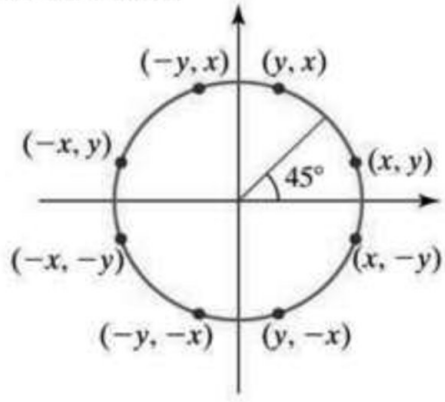
Implement Bresenham's algorithm to draw circle.

VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

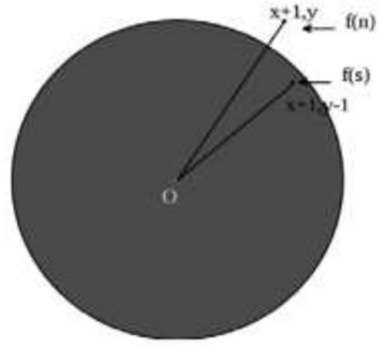
The equation of circle is $x^2+y^2=r^2$, where (x, y) are coordinates of a center and r is radius. To draw circle in computer graphics we use Bresenham's circle drawing algorithm and Mid Point circle drawing algorithm.



It's not easy to display a continuous arc on the raster display. Instead, we have to choose the nearest pixel position to complete the arc.

Bresenham's Circle algorithm:

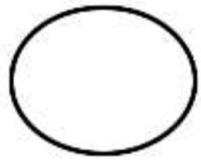
Bresenham's algorithm is based on the idea of determining the subsequent pixels required to draw the circle. x+1,y is to fins the next pixel to draw the circle.



Procedure:

1. Set initial values of center of the circle (xc, yc) and (x, y)
2. Set decision parameter d to d = 3 – (2 * r).
3. Repeat steps 4 to 8 until x < = y
4. Call drawCircle(int xc, int yc, int x, int y) function.
5. Increment value of x.
6. If d < 0, set d = d + (4*x) + 6
7. Else, set d = d + 4 * (x – y) + 10 and decrement y by 1.
8. Call drawCircle(int xc, int yc, int x, int y) function.

Sample Output of the source example:



VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write equation of a circle.
2. Write the algorithm to draw 8-way symmetry of a circle.
3. How the value of decision parameter (d) is calculated.

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Calculate pixels for a circle with radius 10 using Bresenham's algorithm.

d	x	y	Plot(x,y)

2. Draw a circle with center (50,50) and radius 20 by using Bresenham's algorithm.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2. http://www.vrarchitect.net/anu/cg/Circle/symmetry8.en.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

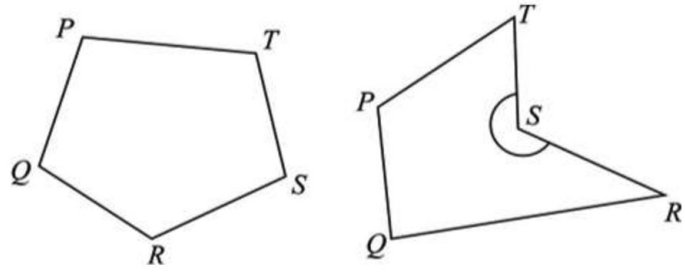
1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.5: Program to fill polygon.

I. Practical Significance

Polygon is a chain of connected line segments. For filling polygons with particular colors, you need to determine the pixels falling on the border of the polygon and those which fall inside the polygon.



II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.

V. Practical Outcome (POs):

Write a program to fill Polygon using Flood fill algorithm.

VI. Relevant Affective domain related Outcome(s)

a. Experiment with graphics environment.
b. Follow safety/ethical practices.
c. Maintain tools and equipment.

VII. Minimum Theoretical Background

Flood Fill Algorithm

In Flood Fill algorithm we start with some seed and examine the neighboring pixels, however pixels are checked for a specified interior color instead of boundary color and is replaced by a new color. It can be done using 4 connected or 8 connected region method.

Procedure:

Flood-fill (node, target-color, replacement-color):
1. If target-color is equal to replacement-color, return.
2. If the color of node is not equal to target-color, return.
3. Set the color of node to replacement-color.
4. Perform Flood-fill (one step to the south of node, target-color, replacement-color).
Perform Flood-fill (one step to the north of node, target-color, replacement-color).
Perform Flood-fill (one step to the west of node, target-color, replacement-color).
Perform Flood-fill (one step to the east of node, target-color, replacement-color).
5. Return.

Example :

```
circle(100,100,50);
floodfill(100,100,RED);
```

Fills the circle with red colour.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XI. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety practices.

XII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIII. Result (Output of the Program)

.....

.....

.....

XIV. Conclusion(s)

.....

.....

.....

XV. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Define polygon.
2. Explain types of polygon.

3. List Coordinates of neighboring pixels in 8-connected method for seed pixel with coordinates(x,y)
4. List Coordinates of neighboring pixels in 4-connected method for seed pixel with coordinates(x,y)
5. Explain inside-outside test of polygon.

(Space for Answers)

XVI. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. WAP to draw hexagon and fill hexagon with pink color using flood fill algorithm with 8-connected method.
ii. WAP to draw triangle (use line function) and fill it with blue color using flood fill algorithm with 4-connected method.

(Space for Answers)

XVII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XVIII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

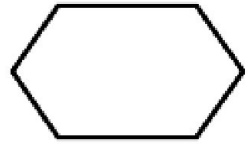
1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.6: Program to fill polygon using boundary fill algorithm.

I. Practical Significance

Polygon is a chain of connected line segments in a close loop. For filling polygons with particular colors, you need to determine the pixels falling on the border of the polygon and those which fall inside the polygon.



II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.

V. Practical Outcome (POs):

Write a program to fill Polygon using 4-connected region or 8-connected region method for Boundary fill algorithm.

VI. Relevant Affective domain related Outcome(s)

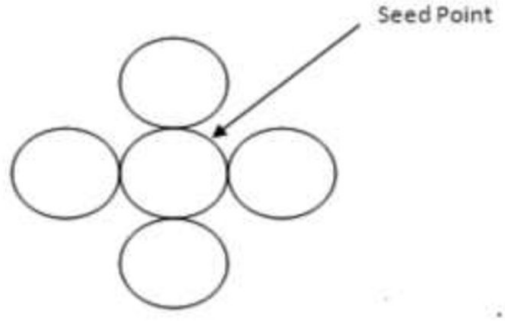
a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety practices.

VII. Minimum Theoretical Background

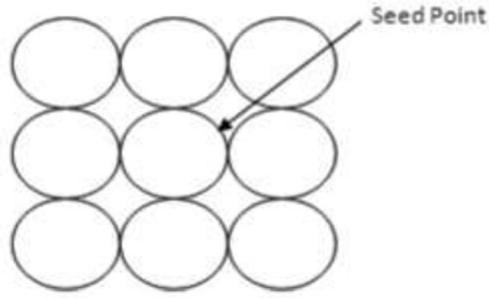
Boundary Fill Algorithm

Boundary fill algorithm picks a point inside that is a seed point of an object and starts to fill until it hits the boundary of the object. If boundary pixels are not reached, pixels are highlighted and the process is continued until boundary pixels are reached. The color of the boundary and the color that we fill should be different for this algorithm to work.

In this algorithm, we assume that color of the boundary is same for the entire object. The boundary fill algorithm can be implemented by 4-connected pixels or 8-connected pixels.



4 – Connected



8 – Connected

Following is a Boundary fill function module to be included in the algorithm for filling "4-connected" region with colour specified in parameter fill colour as *f_colour* up to a boundary colour specified with parameter boundary colour as *b_colour*.

4 – Connected Boundary Fill

```
boundary_ fill_fun (x, y, f_color, b_color)
{
If ( getpixel (x,y) != b_colour && getpixel (x,y) != f_colour)
putpixel(x,y,f_colour)
boundary_fill_ fun ( x+1, y, f_colour, b_colour);
boundary_fill_ fun ( x, y+1, f_colour, b_colour);
boundary_fill_ fun ( x-1, y, f_colour, b_colour);
boundary_fill_ fun ( x, y-1, f_colour, b_colour);
}
```

NOTE: It is assumed that 'getpixel' is standard library function which gives colour of specified pixel and 'putpixel' draws the pixel point with specified colour.

8 – Connected Boundary Fill:

To enhance speed of filling colour one may look for alternative of 4 – connected as 8-connected. In this algorithm all adjacent pixels will be consider for filling till the match is true. Algorithm is as follows.

```
boundary_fill(x, y, f_colour, b_colour)
{
   if(getpixel(x, y)! = b_colour && getpixel(x, y)! = f_colour)
   {
       putpixel(x, y, f_colour);
       boundary_fill(x + 1, y, f_colour, b_colour);
       boundary_fill(x - 1, y, f_colour, b_colour);
       boundary_fill(x, y + 1, f_colour, b_colour);
       boundary_fill(x, y - 1, f_colour, b_colour);
       boundary_fill(x + 1, y + 1, f_colour, b_colour);
       boundary_fill (x - 1, y - 1, f_colour, b_colour);
       boundary_fill(x + 1, y - 1, f_colour, b_colour);
       boundary_fill(x - 1, y + 1, f_colour, b_colour);
    }
}
```

Procedure:

Step 1 – Initialize the value of seed point seedx, seedy, fcolor and dcol.

Step 2 – Define the boundary values of the polygon.

Step 3 – Check if the current seed point is of default color, then repeat the steps 4 and 5 till the boundary pixels reached.

Step 4 – Change the default color with the fill color at the seed point.

Step 5 – Recursively follow the procedure with four neighborhood points.

Step 6 – Exit

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

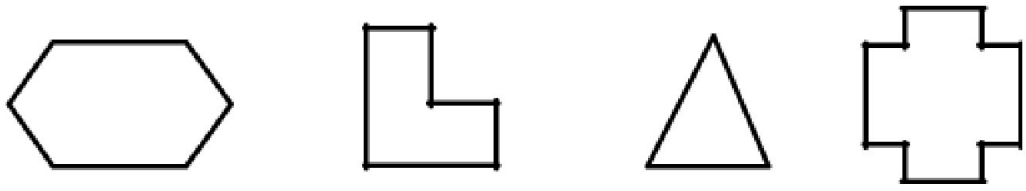
XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Compare 4-connected and 8-connected method to fill polygon.

2. Identify type of polygon in following diagrams.



3. Give the basic difference between flood fill and boundary fill.

(Space for Answers)

XVII. Exercise

Attempt Q1. and teacher shall allot Q. i/ Q.ii from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Write a program to draw pentagon and fill it with red color using boundary fill algorithm with 8-connected method.
ii. Write a program to draw trapezoid and fill it with blue color using boundary fill algorithm with 4-connected method.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 7: Program for two dimensional transformations (translation and scaling).

I. Practical Significance

One of the most common and important tasks in Computer graphics is to transform the coordinates (position, orientation, and size) of objects. Transformations are one of the primary vehicles used in Computer graphics to manipulate objects in two or three-dimensional space. There are different types of transformations such as translation, scaling up or down, rotation, shearing, etc. When a transformation takes place on a 2D plane, it is called 2D transformation.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs for 2-D and 3-D transformations.

V. Practical Outcome (POs):

Write a program for two dimensional transformations.

i) Translation

ii) Scaling

VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.

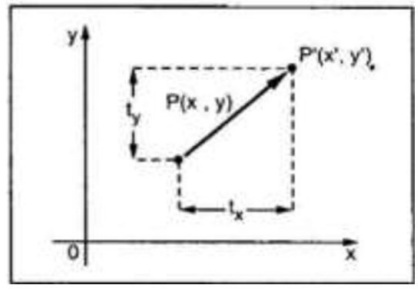
b. Experiment with graphics environment.

c. Follow safety practices.

VII. Minimum Theoretical Background

Translation:

Giving linear displacement to an object along X and Y axis in a particular direction in a plane is translation. A translation is a process of changing the position of an object in a straight line path from one coordinate location to another. We can translate a point in 2D by adding translation coordinate (t_x, t_y) to the original coordinate (X, Y) to get the new coordinate (X', Y').



By adding translation vector ,

$$X' = X + t_x$$

$$Y' = Y + t_y$$

The pair (t_x, t_y) is called the translation vector or shift vector. The above equations can also be represented using the column vectors.

$$P = \begin{bmatrix} X \\ Y \end{bmatrix} \qquad P' = \begin{bmatrix} X' \\ Y' \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

We can write it as –

$$P' = P + T$$

Scaling:

Scaling is a process that increases and decreases original size of an object by virtue of it the object can be made big or small. This transformation makes necessary changes in the size of an object keeping its original shape in tact. The operation can be carried out for an object by multiplying the coordinate value(x, y) of each vertex by scaling factors Sx and Sy to produce the transformed coordinates (x`, y`). Only condition is to ensure that the base point is remained unaltered. Changing the size of an object is a scaling transformation. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result.

Let us assume that the original coordinates are (X, Y), the scaling factors are (S_X, S_Y), and the produced coordinates are (X', Y'). This can be mathematically represented as shown below –

$$\mathbf{X' = X \cdot S_X \text{ and } Y' = Y \cdot S_Y}$$

The scaling factor S_X, S_Y scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below –

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

OR

$$\mathbf{P' = P \cdot S}$$

Where S is the scaling matrix.

The scaling is shown in the following figure.

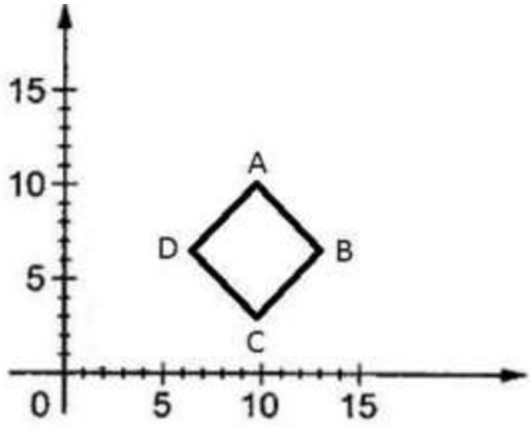


Figure: Scaling before

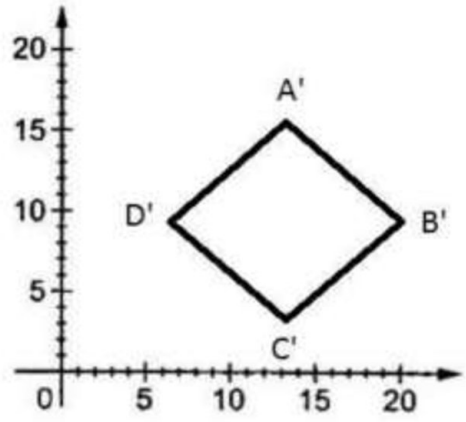


Figure: Scaling after

Procedure:

Step 1: Start the program.
Step 2: Input the object coordinates
Step 3: For Translation
  a) Enter the translation factors tx and ty.
  b) Move the original coordinate position (x,y) to a new position (x1,y1).ie. x=x+x1, y=y+y1.
  c) Display the object after translation
Step 4: For Scaling
  a) Input the scaled factors sx and sy.
  b) The transformed coordinates (x1,y1) , x1=x.sx and y1=y.sy.
  c) Display the object after scaling
Step 5: Stop the Program.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write the transformation matrix for 2D Translation.
2. Write the transformation matrix for 2D Scaling.
3. What does scaling transformation do?
4. Whether size of object remains same or changed in case of translation?

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVII. Exercise

Attempt Q1. And teacher shall allot Q. 2/Q.3 from the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1.Translate the polygon with co-ordinates A(2,5),B(7,10) and C(10,2) by 3 units in x direction and 4 units in y direction.

2. Scale the polygon with co-ordinates A(2,5),B(7,10) and C(10,2) by 2 units in x direction and 2 units in y direction.

3. Give a 3x3 homogeneous co-ordinate transformation matrix for each of the following translations.

i. Shift the image to the right 3 units
ii. Shift the image up 2 units
iii. Move the image down ½ unit and right 1 unit
iv. Move the image down 2/3 unit and left 4 units.

4. Find the transformation matrix that transforms the given square ABCD to half its size with center still remaining at the same position. The co-ordinates of the square are : A(1,1),B(3,1),C(3,3) and D(1,3) and center at (2,2).Also find the resultant co-ordinates of square.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related(10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 8: Program for two dimensional transformations (Rotation).

I. Practical Significance

One of the most common and important tasks in Computer graphics is to transform the coordinates (position, orientation, and size) of objects. It is a process by virtue of which the object can rotate to specific angle. A 2-D rotation is applied to an object by repositioning it along a circular path in the x, y plane. To generate a rotation, we give a rotation angle 'θ' and the position of the rotation point about which the object is to be rotated.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs for 2-D and 3-D transformations.

V. Practical Outcome (POs):

Write a program for two dimensional Rotations.

VI. Relevant Affective domain related Outcome(s)

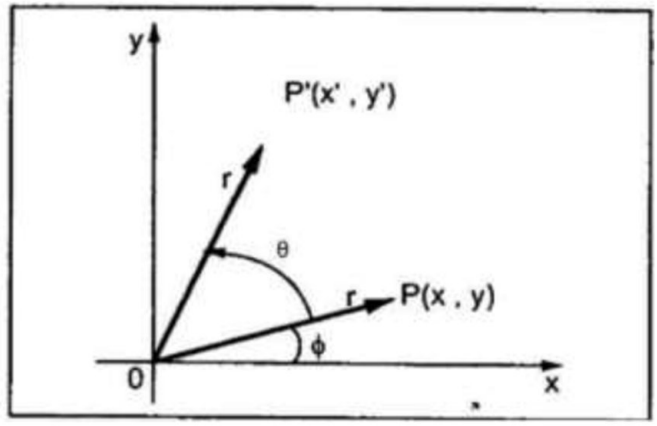
a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety practices.

VII. Minimum Theoretical Background

Rotation:

Giving angular displacement to an object is a rotation. In rotation, we rotate the object at particular angle **θ** (theta) from its origin. From the following figure, we can see that the point P(X, Y) is located at angle φ from the horizontal X coordinate with distance r from the origin.

Consider, you want to rotate it at the angle θ. After rotating it to a new location, you will get a new point P' (X', Y').



The original coordinate of point P(X, Y) can be represented as –

$$X = r\cos\phi \ldots\ldots (1)$$

$$Y = r\sin\phi \ldots\ldots (2)$$

Same way we can represent the point P' (X', Y') as –

$$x' = r\cos(\phi + \theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \ldots\ldots (3)$$

$$y' = r\sin(\phi + \theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \ldots\ldots (4)$$

Substituting equation (1) & (2) in (3) & (4) respectively, we will get transformation equations for rotating point P(x,y) through an angle **θ** about the origin:

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

Representing the above equation in matrix form,

$$[X'Y'] = [XY]\begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} OR$$

$$P' = P \cdot R$$

Where R is the rotation matrix and is represented as

$$R = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}$$

The rotation angle can be positive and negative (i.e. clockwise or counterclockwise) For positive rotation angle, we can use the above rotation matrix. However, for negative angle rotation, the matrix will change as shown below

$$R = \begin{bmatrix} cos(-\theta) & sin(-\theta) \\ -sin(-\theta) & cos(-\theta) \end{bmatrix}$$

$$= \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} (\because cos(-\theta) = cos\theta \; and \; sin(-\theta) = -sin\theta)$$

Procedure:

Step 1: Start the program.
Step 2: Input the object coordinates
Step 3: For Rotation
a) Enter the radian for rotation angle θ.
b) Rotate a point at position (x,y,z) through an angle θ about the origin
x1=xcos θ – ysin θ , y1=ycos θ + xsin θ.
c) Display the object after rotation
Step 4: Stop the Program.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write the transformation matrix for 2D Rotation.
2. Write rotation matrix for a rotation angle 30^0.

(Space for Answers)

XVII. Exercise

Attempt Q1. And teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

A point (4, 3) is rotated counterclockwise by an angle of 45^0. Find the rotation matrix and the resultant point.

2. Perform a counterclockwise 45^0 rotation of triangle A (2, 3), B (5, 5), C (4, 3) about point (1,1).
3. Find a transformation of triangle A(1,0),B(0,1),C(1,1) by
   i. Rotating 45^0about the origin and then translating one unit in x and y direction.
   ii. Translating one unit in x and y direction and then rotating 45^0about the origin.
4. Consider the square A(1,0),B(0,0),C(0,1),D(1,1).Rotate the square by 45^0 anticlockwise direction.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 9: Program for two dimensional transformations (Reflection and Shearing)

I. Practical Significance

One of the most common and important tasks in Computer graphics is to transform the coordinates (position, orientation, and size) of objects. It is a transformation which slants or bends an object to specified direction. There are two types of shearing transformation available in Computer graphics. One which slants x coordinate values is known as X shearing and one that slants y coordinate values is known as Y shearing. Irrespective of shearing only one co-ordinate is change its coordinate and other values are same. In Reflection the mirror copy of an object is generated based on the axis to which reflection is to be made.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you.

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs for 2-D and 3-D transformations.

V. Practical Outcome (POs):

Write a program for transformation and shearing.

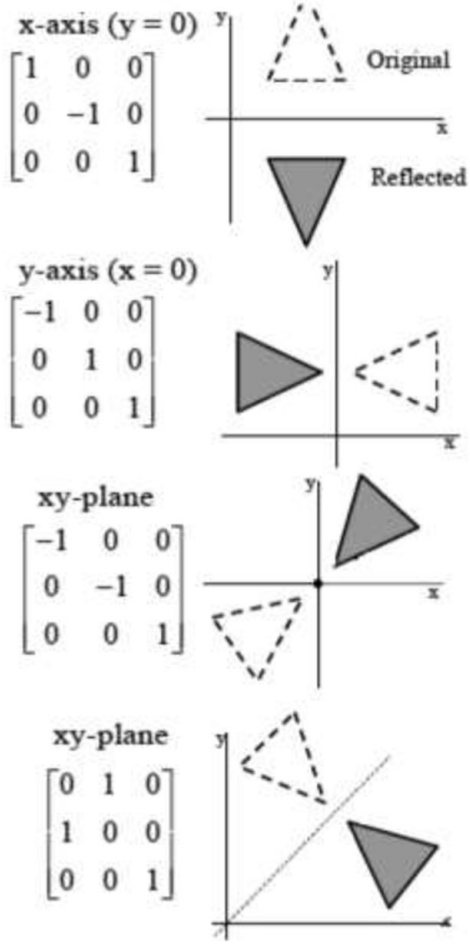
VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

1. Reflection:-

Reflection is a transformation that results into a mirror image of original object. In reflection transformation, the size of the object does not change. The mirror image of any image for 2D reflection is generated with respect to the "Axis of Reflection". For that we need to rotate main object 180 Degrees about the reflection axis. Reflection transformation is generally implemented with respect to the coordinate axes or its coordinate origin as the scaling transformation with t minus (negative) scaling factors.

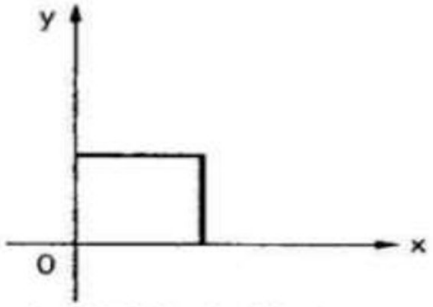


2. Shear:

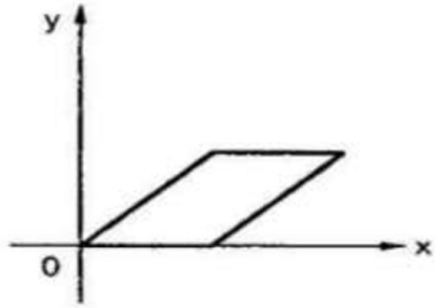
Slanting the shape of an object is shearing transformation. There are two shear transformations X-Shear and Y-Shear. One shifts X coordinates values and other shifts Y coordinate values. In fact, in both the cases only one coordinate changes its coordinates and other preserves its values. Shearing is also called as Skewing.

X-Shear

The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.



(a) Original object



(b) Object after x shear

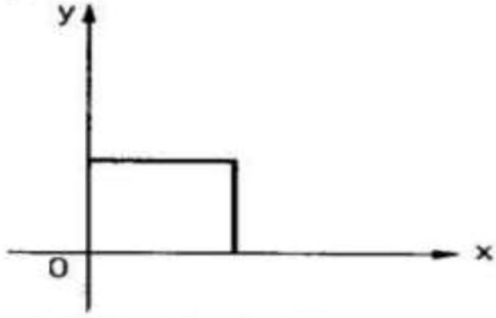
The transformation matrix for X-Shear can be represented as –

$$\begin{pmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

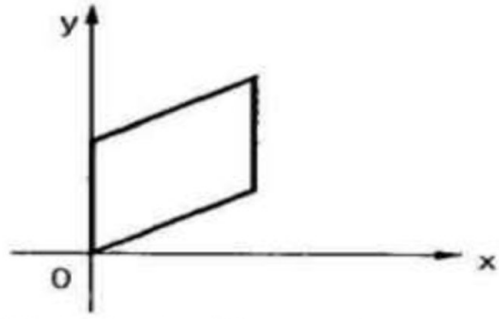
$$X' = X + Sh_x \cdot Y$$
$$Y' = Y$$

Y-Shear

The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.



(a) Original object



(b) Object after y shear

The Y-Shear can be represented in matrix from as –

$$\begin{pmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Y' = Y + Sh_y \cdot X$$
$$X' = X$$

Procedure :

Step 1: Start the program.

Step 2: Input the object coordinates

Step 3: For Shearing

a) Input the shearing factors shx and shy.

b) Shearing related to x axis: Transform coordinates x1=x+shx*y and y1=y.

c) Shearing related to y axis: Transform coordinates x1=x and y1=y+shy*x.

d) Input the xref and yref values.

e) X axis shear related to the reference line y-yref is x1=x+shx(y-yref) and y1=y.

f) Y axis shear related to the reference line x=xref is x1=x

g) Display the object after shearing

Step 4: For Reflection

Reflection can be performed about x axis and y axis.

a) Reflection about x axis: The transformed coordinates are x1=a and y1=-y.

b) Reflection about y axis: The transformed coordinates are x1=x and y1=y.

c) Display the object after reflection

Step 5: Stop the Program.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write the transformation matrix for 2D Reflection.
2. Write the transformation matrix for 2D Shear.
3. Differentiate between X-shear and Y-shear.
4. Define Reflection and Shearing.

(Space for Answers)

XVII. Exercise

Attempt Q1. And teacher shall allot Q. 1/Q.2 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. A point (4, 3) is rotated counterclockwise by an angle of 45^0. Find the rotation matrix and the resultant point. Apply the Shearing transformation to square with A(0,0),B(1,0),C(1,1) and D(0,1) as given below :
   i. Shear parameter value of 0.5 relative to the line Yref= -1;
   ii. Shear parameter value of 0.5 relative to the line Xref= -1;

2. Apply shearing transformation to square with A(0,0),B(1,0),C(1,1) and D(0,1).If Sh_x =0.5 then find the resultant co-ordinates.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100% (25)**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……...

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 10: Program for three dimensional transformations (Translation and Scaling)

I. Practical Significance

3D Computer graphics are graphics that use a three dimensional representation of geometric data that is stored in the Computer for the purposes of performing calculations and rendering 2D images. 3D transformations are extended from 2D methods by including considerations for the Z coordinate.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you.

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs for 2-D and 3-D transformations.

V. Practical Outcome (POs):

Write a program for three dimensional Translation and Scaling.

VI. Relevant Affective domain related Outcome(s)

a. Plan, construct, compile, debug and test programs.
b. Experiment with graphics environment.
c. Follow safety practices.

VII. Minimum Theoretical Background

Transformations in 3D:-

As, 2D transformations, it has a coordinate system with three axes as a basis. In this learning material all reasoning in space is done in a right hand system.

This means that if I put my right hand vertically down, with my fingers along the positive x-axis, and bend the hand towards the y-axis, the thumb will point up along the positive z-axis.

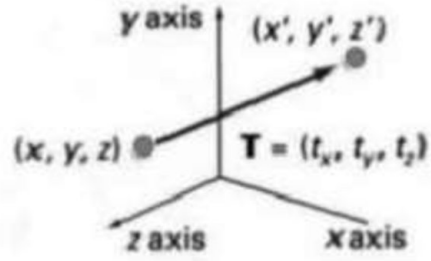
We use homogeneous coordinates from the beginning. This means that the general transformation matrix is a 4x4 matrix, and that the general vector form is a column vector with four rows.

$P_2=M\cdot P_1$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

1. Translation:

Moving of object is called translation.



A translation in space is described by t_x, t_y and t_z. It is easy to see that this matrix realizes the equations:

$$P'=T.P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

This matrix representation is equivalent to following three equations.

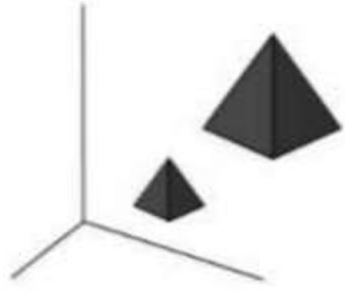
$$X' = X + t_x$$

$$Y' = Y + t_y$$

$$Z' = Z + t_z$$

2. Scaling:

Changing the size of an object is called scaling transformation. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result. The following figure shows the effect of 3D scaling −



In 3D scaling operation, three coordinates are used. Consider that, the original coordinates are (X, Y, Z), scaling factors are (S_X, S_Y, S_Z) respectively, and the produced coordinates are (X', Y', Z'). This can be mathematically represented in matrix form as below –

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot S$$

$$[X' \;\; Y' \;\; Z' \;\; 1] = [X \;\; Y \;\; Z \;\; 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [X.S_x \;\; Y.S_y \;\; Z.S_z \;\; 1]$$

Scaling in space is described by S_x, S_y and S_z. We see that this matrix realizes the following equations:

$$X' = X \cdot S_x$$
$$Y' = Y \cdot S_y$$
$$Z' = Z \cdot S_z$$

Procedure:

Step 1: Start the program.
Step 2: Input the object coordinates
Step 3: For Translation
a) Enter the translation factors t_x, t_y and t_z.
b) Move the original coordinate position (x,y,z) to a new position (x1,y1,y1).ie. x=x+x1, y=y+y1 and z=z+z1.
c) Display the object after translation
Step 4: For Scaling
a) Input the scaled factors sx,sy and sz.
b) The transformed coordinates (x1,y1,z1) , $x1=x.s_x$, $y1=y.s_y$ and $z1=z.s_x$.
c) Display the object after scaling
Step 5: Stop the Program.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write the transformation matrix for 3D Translation.
2. Write the transformation matrix for 3D Scaling.
3. What Z- Coordinate indicates in 3D transformations.
4. Explain Homogeneous coordinates.

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVII. Exercise

Attempt Q1. And teacher shall allot Q. 1/Q.1 from the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to draw a cube in C by using 'bar3d' function. Translate the cube by 75 units in X, 75 units in Y and 75 units in Z direction.

2. Write a program to draw a cube in C by using 'bar3d' function. Scale it to double of its original size.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No. 11: Program for three dimensional transformations (Rotation)

I. Practical Significance

3D Computer graphics are graphics that use a three dimensional representation of geometric data that is stored in the Computer for the purposes of performing calculations and rendering 2D images. To emphasize working with the real world problem statements and games 3D transformations are used.

Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs for 2-D and 3-D transformations.

V. Practical Outcome (POs):

Write a program for three dimensional Rotation.

VI. Relevant Affective domain related Outcome(s)

a. Plan, construct, compile, debug and test programs.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Rotation:-

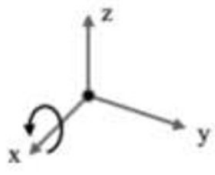
3D rotation is not same as 2D rotation. In 3D rotation, we have to specify the angle of rotation along with the axis of rotation. We can perform 3D rotation about X, Y, and Z axis. They are represented in the matrix form as below –

$$R_x(\theta)=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_y(\theta)=\begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta)=\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The following figure explains the rotation about various axes –

X-axis rotation-



Y-axis rotation-

Z-axis rotation-

Procedure:

Step 1: Start the program.
Step 2: Input the object coordinates
Step 3: For Rotation
a) Enter the radian for rotation angle θ.
b) Perform rotation about each axis.

i) Z-Axis Rotation

Z-axis rotation is identical to the 2D case:

$x' = x*\cos\theta - y*\sin\theta$

$y' = x*\sin\theta + y*\cos\theta$

$z' = z$

ii) X-Axis Rotation

X-axis rotation looks like Z-axis rotation if replace:

$y' = y*\cos\theta - z*\sin\theta$

$z' = y*\sin\theta + z*\cos\theta$

$x' = x$

iii) Y-Axis Rotation

Y-axis rotation looks like Z-axis rotation if replace:

$z' = z*\cos\theta - x*\sin\theta$

$x' = z*\sin\theta + x*\cos\theta$

$y' = y$

c) Display the object after rotation

Step 4: Stop the Program.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write the transformation matrix for 3D Rotation about Z-axis.
2. Write the transformation matrix for 3D Rotation about X-axis.

3. Write the transformation matrix for 3D Rotation about Y-axis.

(Space for Answers)

XVII. Exercise

Attempt Q1. And teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. WAP to draw a cube in C by using 'bar3d' function. Rotate the cube by 45^0 around X-axis.
2. WAP to draw the triangle defined by three vertices A(0,2,1), B(2,3,0), C(1,2,1). Rotate the given triangle around Y-axis.

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. .....
2. .....
3. .....
4. .....

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.12: Program to clip line using cohen sutherland line clipping algorithm

I. Practical Significance

The process that identifies which portion is inside and which portion is outside the specified region of space is called as a clipping. The Cohen–Sutherland algorithm is a line clipping algorithm which uses four digit code to point out which of nine regions contain the end point of the line. The four bit code is called as region code.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer Programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Implement various clipping algorithms using C.

V. Practical Outcome (POs):

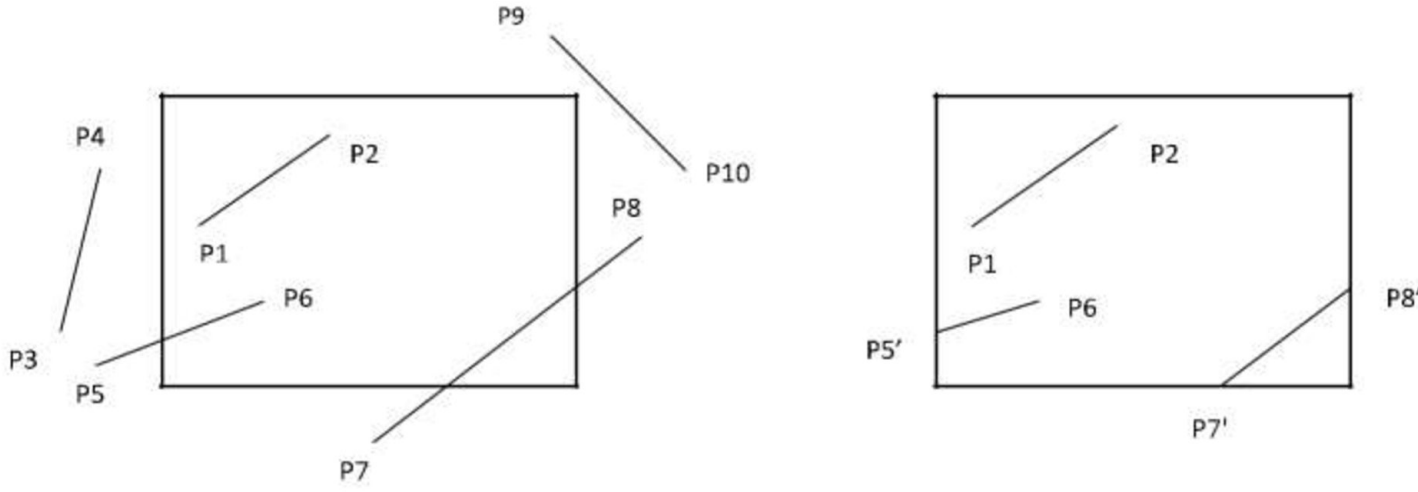
Write a program to clip line using Cohen - Sutherland line clipping algorithm.

VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Line clipping algorithm:
It is the process of removing lines or portions of lines outside of an area called clipping region or window. Any line or the part of any line which is outside the clipping window is removed. Following figure shows the lines before and after clipping the window.



Before clipping **After Clipping**

Lines fall into one of the following clipping categories:

a. **Completely Visible:** These lines are said to be interior to the clipping window. Both end points of these lines are interior to window. In the above figure line P1P2 is completely visible.

b. **Completely Invisible:** If both end points of a line are outside the window and not crossing the boundaries the line is completely invisible in window. In above figure P3P4, P9P10 are completely invisible.

c. **Partially Visible:** If the lines cross the clipping boundaries the lines are partially visible. Line P5P6, P7P8 is partially visible.

Four bit codes (Region Codes):
To find the clipping category of the line, four bit code is used. The codes for the end points of the lines are assigned by considering the region in which the end points resides.

1001	1000	1010
0001	0000 Clipping Region	0010
0101	0100	0110

Each bit in the region code is used to indicate one of the four relative co-ordinates positions of point with respect to clipping window to left, right, top and bottom.

Four bit is assigned to points as,

Top	Bottom	Right	Left
4	3	2	1

i. Bit 1 is set to 1 if point lies to the left of the window.

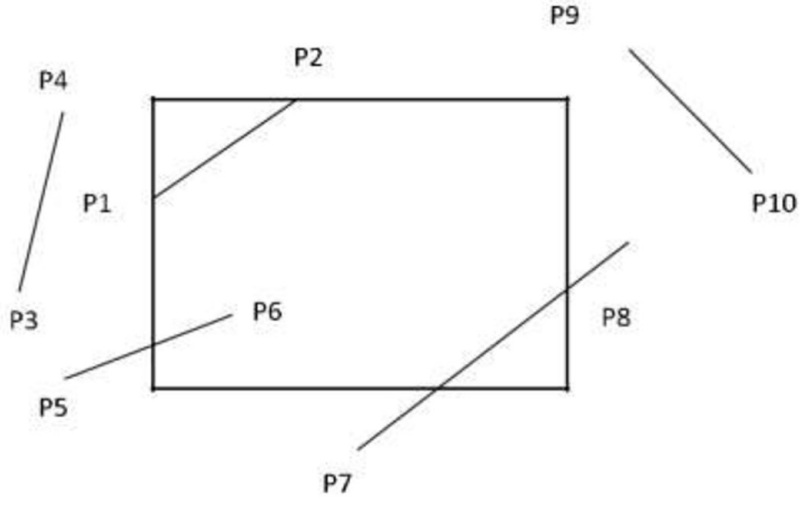
ii. Bit 2 is set to 1 if point lies to the right of the window.

iii. Bit 3 is set to 1 if point lies to the below of the window.

iv. Bit 4 is set to 1 if point lies to the above of the window.

v. Otherwise the bit is set to zero.

Example:



Four bit code for the end points are,

P1	0000	P6	0000
P2	0000	P7	0100
P3	0001	P8	0010
P4	0001	P9	1000
P5	0001	P10	0010

Identifying category of line using four bit code:

From the four bit codes of all line end points clipping categories of lines are determined as below,

1. If four bit codes of both endpoints of the line is 0000 then the line is completely visible.
2. If bitwise logical AND of four bit codes of both endpoints is not 0000 then the line is completely invisible.
3. If bitwise logical AND of four bit codes of both endpoints is 0000 then the line is partially visible and needs clipping.

To find out intersection boundary:

Choose the end points whose region code is not 0000. Then depending upon the position of the 1 in region code following scheme decides with which boundary line intersects.

1. If bit 1 is 1, intersects with line $Y=Y_{max}$ (top boundary).
2. If bit 2 is 1, intersects with line $Y=Y_{min}$ (bottom boundary).
3. If bit 3 is 1, intersects with line $X=X_{max}$ (right boundary).
4. If bit 4 is 1, intersects with line $X=X_{min}$ (left boundary).

To find out co-ordinates of intersection point:

Once line of intersection is known then point of intersection can be calculated as,

- If boundary line is vertical, then

  $X_i=X_{min}$ or X_{max}

  $Y_i=Y_1+m(X_i-X_1)$

- If boundary line is Horizontal, then

  $X_i=X_1+(Y_i-Y_1)/m$

Where, m is slope of line = $\frac{Y_2-Y_1}{X_2-X_1}$

(X_1,Y_1) = end point of line

(X_i, Y_i)= intersection point

Now replace end points (X_1, Y_1) with intersection point (X_i, Y_i) and display this new line joining two intersection points or a line joining one intersection point and other end point in 0000 region code.

Procedure:

1. Given a line segment with endpoint $P_1=(X_1,Y_1)$ and $P_2=(X_2,Y_2)$
2. Compute the 4-bit codes for each endpoint.

If both codes are **0000**,(bitwise OR of the codes yields 0000) line lies completely **inside** the window: pass the endpoints to the draw routine.
If both codes have a 1 in the same bit position (bitwise AND of the codes is **not** 0000), the line lies **outside** the window. It can be trivially rejected.

3. If a line cannot be trivially accepted or rejected, at least one of the two endpoints must lie outside the window and the line segment crosses a window edge. This line must be **clipped** at the window edge before being passed to the drawing routine.
4. Examine one of the endpoints, say $P_1=(X_1,Y_1)$. Read P_1's 4-bit code in order: **Left**-to-**Right**, **Bottom**-to-**Top**.
5. When a set bit (1) is found, compute the intersection **I** of the corresponding window edge with the line from P_1 to P_2. Replace P_1 with **I** and repeat the procedure.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Define clipping.

ii. How to calculate intersection points of a line if line is partially visible.

iii. Give 3 possible conditions to clip line using Cohen Sutherland line clipping algorithm.

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

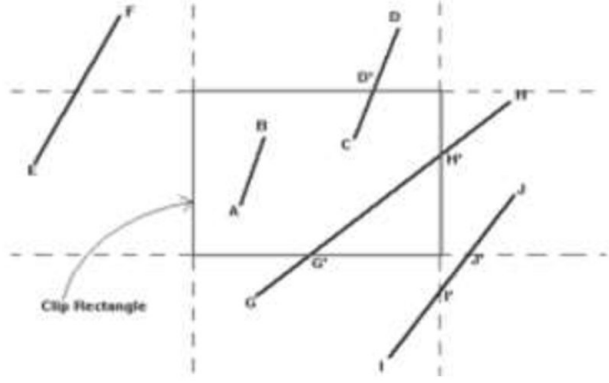
.....

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write four bit code for following lines and determine clipping categories of each line.



(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related(10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.13: Program to clip line using midpoint subdivision line clipping algorithm.

I. Practical Significance

Clipping process removes objects, lines, or parts of objects that are outside the viewing pane. The strength of this algorithm over the Cohen-Sutherland algorithm is that it requires no floating point arithmetic to find the point of intersection with the line and the clip boundary. The midpoint subdivision algorithm clips a line by finding the endpoints of the visible portion of the line segment. Each endpoint can be found by an identical process and given appropriate hardware; this can be done in parallel for both endpoints.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Implement various clipping algorithms using C.

V. Practical Outcome (POs):

Write a program to clip line using Midpoint Subdivision line clipping algorithm.

VI. Relevant Affective domain related Outcome(s)

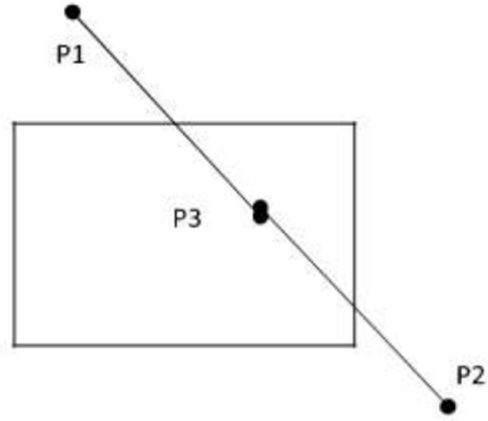
a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Midpoint Subdivision Line clipping algorithm

Midpoint subdivision algorithm works same as Cohen-Sutherland algorithm. It assigns four bit codes at end point of line to find out clipping category of line like completely visible, completely invisible and partially visible.

Lines with partially visible category send for clipping.
Demonstration of midpoint subdivision algorithm for clipping line P1P2:

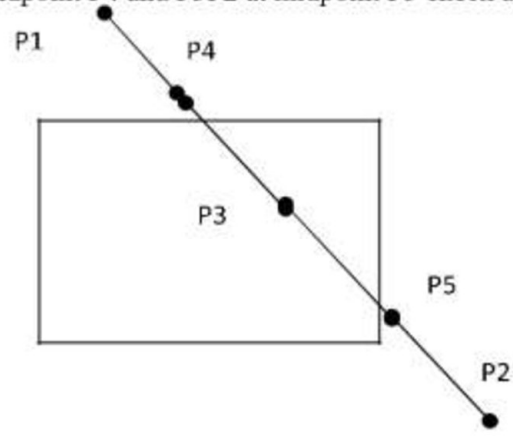


1) Four digit for points P1 and P2 are 1000 and 0010 respectively.

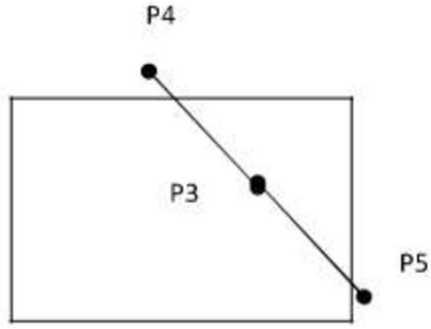
2) As ANDing of four digit codes is 0000 line P1P2 is partially visible.

3) Divide line P1P2 at mid-point P3 forming two segments P1P3 and P3P2 , apply visibility test on both segments. Both lines are in partially visible category and hence we have to again subdivide it.

4) Divide line P1P3 at midpoint P4 and P3P2 at midpoint P5 check all line segments for their visibility.



5) Line P1P4 and line P2P5 are completely invisible and hence removed. Line P4P3 and Line P3P5 are partially visible. Line P4P3 and P3P5 are subdivided again and this procedure continues until all the visible and invisible lines are found.



If P1(x1,y1) and P2(x2,y2) are two points then midpoint P3(x3,y3) is calculated as

x3=(x1+x2)/2

y3=(y1+y2)/2

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	CComputer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Calculate coordinates of midpoint between two points p1(-10,20)andp2(50,10).
ii. Compare Cohen Sutherland and Midpoint subdivision line clipping algorithm.
iii. Write disadvantages of Midpoint subdivision line clipping algorithm.

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

(**Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Using Midpoint subdivision line clipping algorithm illustrate a clipping of a line segment joining two end points A(-1,5),B(3,8) by considering clipping window with left corner at(-3,1) and upper right corner at(2,6).

(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……...

			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.14: Program to clip polygon using sutherland hodgeman polyon clipping algorithm.

I. Practical Significance

The Sutherland-Hodgeman clipping algorithm finds the polygon that is the intersection between an arbitrary polygon (subject polygon) and a convex polygon (clip polygon).
It is used in Computer graphics to reduce the complexity of a scene being displayed by eliminating parts of a polygon that do not need to be displayed.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Implement various clipping algorithms using C.

V. Practical Outcome (POs):

Write a program to clip line using Sutherland Hodgeman Polygon clipping algorithm.

VI. Relevant Affective domain related Outcome(s)

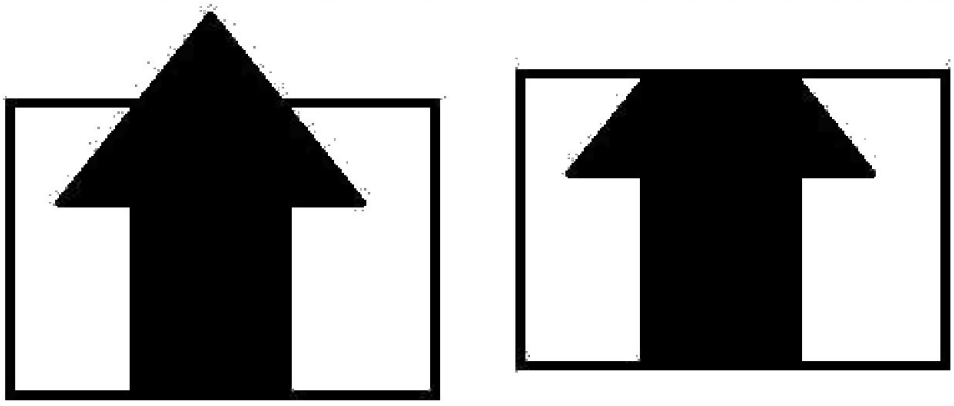
a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Polygon Clipping:

Polygon is specified by set of three or more co-ordinates positions, called vertices. These vertices are connected in sequence by straight line segment. These straight line segments are called as edges or side if the polygon.

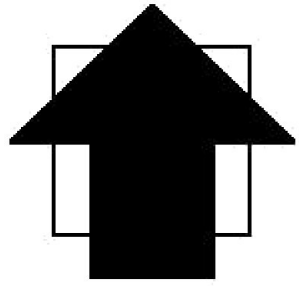
Figure shows original polygon with clipping window and polygon after clipping.



Sutherland - Hodgeman algorithm

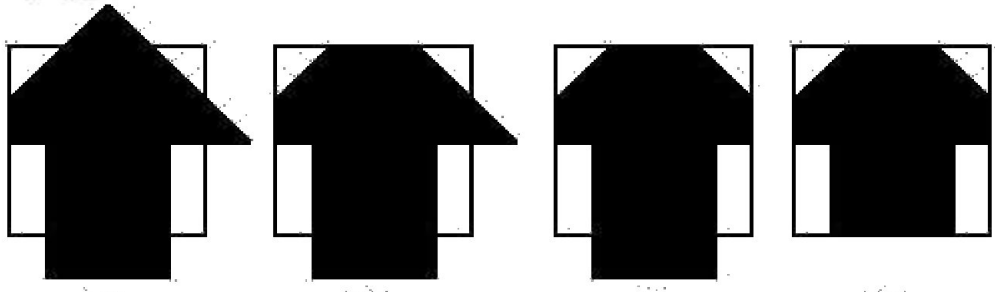
The Sutherland - Hodgeman algorithm performs a clipping of a polygon against each window edge in turn. It accepts an ordered sequence of vertices v1, v2, v3, ..., vn and puts out a set of vertices defining the clipped polygon.

Following figure represents the polygon before clipping has occurred.



Before clipping

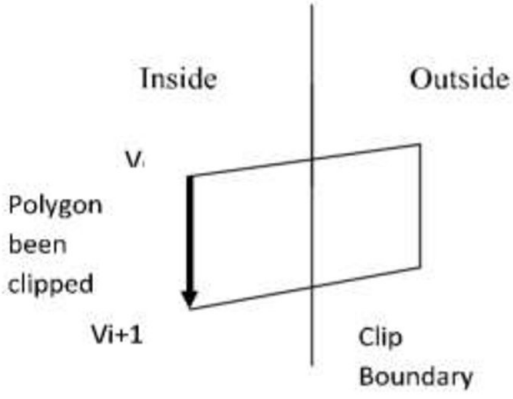
The following figures shows how this algorithm works at each edge, clipping the polygon.



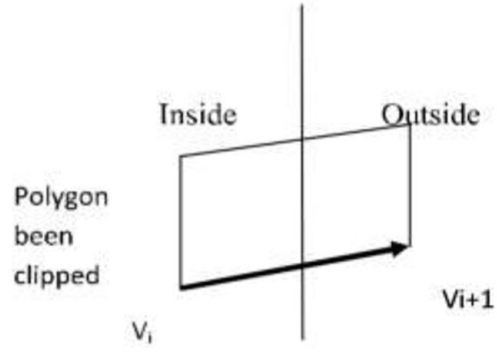
[a][[b] [c] [d]

a. Clipping against the left side of the clip window.
b. Clipping against the top side of the clip window.
c. Clipping against the right side of the clip window.
d. Clipping against the bottom side of the clip window.

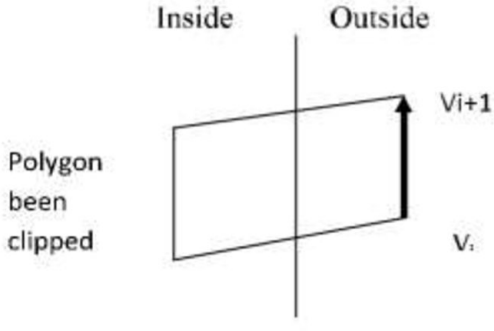
Four test cases of edges:
Every edge of polygon is compared with clipping plane to find out the new vertices called output vertices. According to this there are fur relationships between the edge and clipping boundary.
Case1: If both input vertices are inside the clipping window boundary then add only second vertex to output vertex list.
Case2: If the first vertex is inside the window boundary and second vertex is outside then only intersection with window boundary is added to output vertex list.
Case3: If both input vertices are outside the clipping window boundary then nothing is added to output vertex list.
Case4: If the first vertex is outside the clipping window the clipping window boundary and second vertex is inside it, then both the intersection point of the polygon edge with window boundary and second vertex are added to output vertex list.



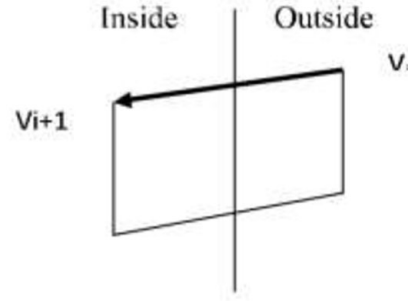
Case: 1



Case: 2



Case: 3



Case: 4

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion (s)

.....

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. If the first vertex is outside the clipping window and second point is inside the clipping window, then write which points are added to output vertex list.

ii. Write the procedure to clip polygon using Sutherland Hodgeman Polygon Clipping algorithm.

(Space for Answers)

.....

.....

.....

.....

.....

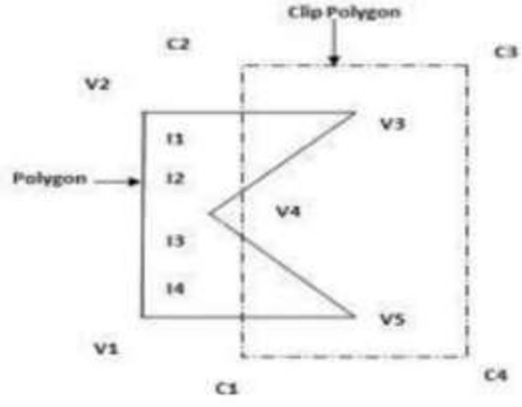
.....

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Clip the following polygon using Sutherland Hodgeman Polygon Clipping algorithm.



(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2. http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related(10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

Marks Obtained			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.15: Program to draw Hilbert's curve.

I. Practical Significance

Objects in real world are not always made up of regular geometric shapes they may includes curves. Drawing curves involves complex mathematical analysis in the form of various interpolation techniques.

II. Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer Programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs in C to create curves and fractals using algorithms.

V. Practical Outcome (POs):

Write a program to draw Hilbert's curve.

VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Hilbert curve

The Hilbert curve is a space filling curve that visits every point in a square grid with a size of 2×2, 4×4, 8×8, 16×16, or any other power of 2. Hilbert curve in Image processing is used for image compression and dithering. It has advantages in those operations where the coherence between neighboring pixels is important. The Hilbert curve is also a special version of a quad tree; any image processing function that benefits from the use of quad trees may also use a Hilbert curve. The Curve can be built by following Successive approximation:

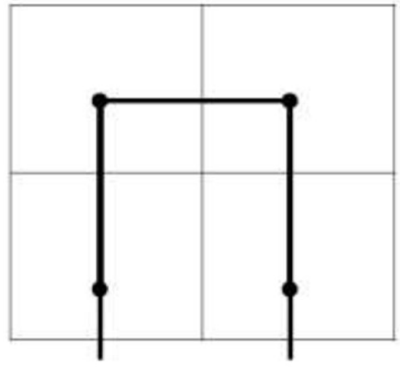


Figure 1: First Approximation to Hilbert Curve

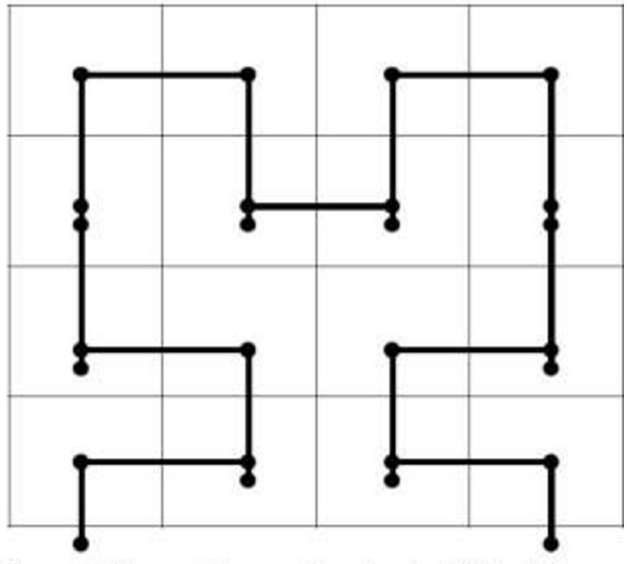


Figure 2: Second Approximation to Hilbert Curve

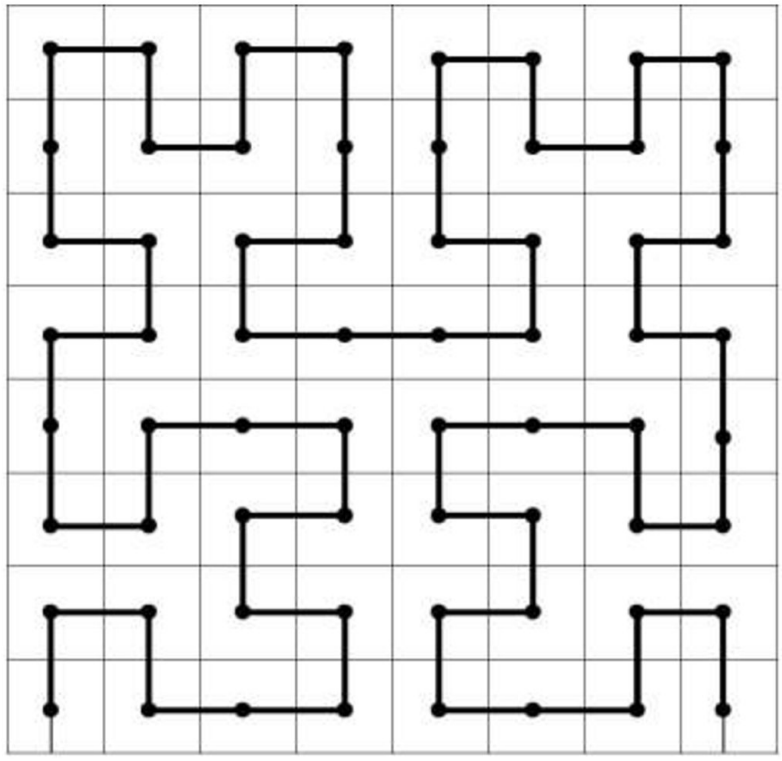


Figure 3: Third Approximation to Hilbert Curve

- The first approximation will divide square into 4 quadrant and draw curve which connect the center point (Figure 1).
- In second again divide each quadrant and again connects the center point (Figure 2).
- In the third approximation again subdivide the quadrant. If again connects the centers of the finest level before stepping to the next level (Figure 3).

Applying the process continuously, remember following things:

1. Curve never cross itself.
2. The curve is arbitrarily close to the every point in the square. There is no limit to subdivisions.
3. The curve fills the square.
4. The length of the curve is infinite, with each subdivision length increase by a factor of 4.
5. There is no limit of length.
6. The constructed curve is topologically equivalent to line Dt=1.
7. The fractal dimensions can be determined as at each subdivision the scale is changed by 2 but the length is changed by 4.
8. For square it takes 4 curves of the half scale to build the full sized object so dimension D can be given as $4 = 2^D$. It must be D=2.
9. The Hilbert curve has topological dimension 1 but fractal dimensions 2.

Procedure:

1. Hilbert subroutine draws the Hilbert curve.
2. It takes as parameters the depth of recursion, and dx and dy values that give the direction in which it should draw.
3. It recursively draws four smaller Hilbert curves and connects them with lines.

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Define Curve.

ii. Write topological and fractal dimension of Hilbert's curve.

(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

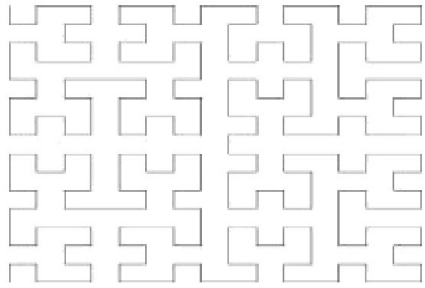
.....

.....

XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Draw following Hilbert's curve.



(Space for Answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related(10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……..

			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

Practical No.16: Program to draw Koch Curve and Bezier Curve.

I. Practical Significance

Objects in real world are not always made up of regular geometric shapes they may includes curves. Drawing curves involves complex mathematical analysis in the form of various interpolation techniques.

Relevant Program Outcomes (POs):

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer Engineering related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Computer Engineering related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Computer Engineering related problems.
- **Engineering tools:** Apply relevant Computer Programming technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

This practical is expect to develop the following skills in you

Develop 'C' programs to draw basic graphics objects:

1. Write syntax for graphics functions.
2. Write and Save a simple C program.
3. Setup graphics drivers, graphics mode and directory to run graphics program.
4. Compile the C program using Turbo C.
5. Debug and execute the program.

IV. Relevant Course Outcome(s):

- Implement standard algorithms to draw various graphics objects using C program.
- Develop programs in C to create curves and fractals using algorithms.

V. Practical Outcome (POs):

Write a program to draw Koch curve and Bezier curve.

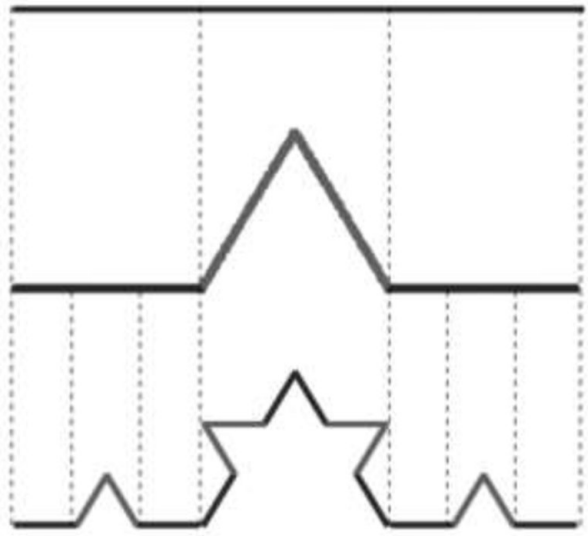
VI. Relevant Affective domain related Outcome(s)

a. Handle command prompt environment.
b. Experiment with graphics environment.
c. Follow safety/ethical practices.

VII. Minimum Theoretical Background

Koch Curve

Koch curve begin with a straight line (the blue segment in the top figure). Divide it into three equal segments and replace the middle segment by the two sides of an equilateral triangle of the same length as the segment being removed (the two red segments in the middle figure). Now repeat, taking each of the four resulting segments, dividing them into three equal parts and replacing each of the middle segments by two sides of an equilateral triangle (the red segments in the bottom figure). Continue this construction.



Suppose repeating the replacements indefinitely, since each repetition increase the length by factor of 4/3, the length of the curve will be infinite but it is folded in a lots of tiny wiggles'.

Its topological dimension is 1, Its fractal dimension is, $4=3^D$.

Solving this fractal dimension gives,

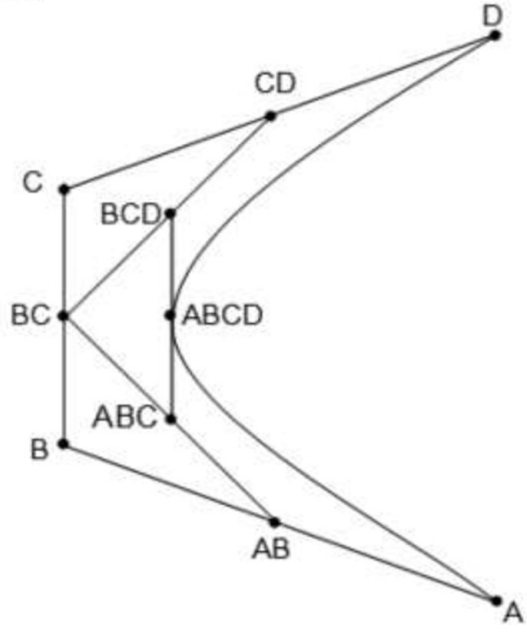
$$D=\log_3 4/\log 3=1.2618$$

Bezier Curves

Bezier Curve is adequate for most graphical applications.

This curve requires four control points. These four control points completely specify the curve.

Additional points cannot be added. We cannot extend Bezier curve but we can take four more points and we can construct a second Bezier curve which can be attached to second Bezier curve.



Procedure:

Koch curve:

The Koch curve is a simple fractal that creates a pretty snowflake-like object. The iteration algorithm is very simple:

1. Start with a straight line:

2. Trisect the line into three segments:

3. Form an equilateral triangle rising out of the middle segment:

4. Repeat, with newly formed segment.

If you start with an equilateral triangle instead of a line, you get the lovely image shown at the top of the article after a few iterations.

Bezier Curve:

To each set of four points P_0, P_1, P_2, P_3 we associate a curve with the following properties:

1. It starts at P_0 and ends at P_3.
2. When it starts from P_0 it heads directly towards P_1, and when it arrives at P_3 it is coming from the direction of P_2.
3. The entire curve is contained in the quadrilateral whose corners are the four given points (their **convex hull**).

VIII. Algorithm

IX. Flow Chart

X. 'C' Program Code

XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards but not limited to	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later with DOSBOX		

XII. Precautions

1. Ensure that all C statements must end with a semicolon (;).
2. Use white spaces in c to describe blanks and tabs.
3. Ensure use of proper graphics function for relevant object.
4. Follow safety/ethical practices.

XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

XIV. Result (Output of the Program)

.....

.....

.....

XV. Conclusion(s)

.....

.....

.....

XVI. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

i. Define fractal.

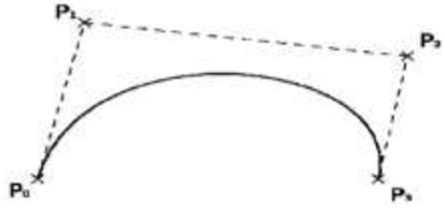
ii. Define fractal dimension and topological dimension.

(Space for Answers)

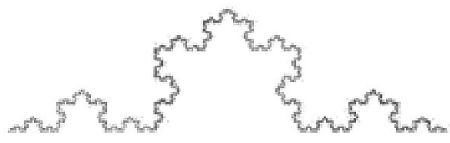
XVII. Exercise

Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:
(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Draw following Bezier curve.



2. Draw following Koch curve.



(Space for Answers)

XVIII. References / Suggestions for further Reading

1. https://books.google.co.in/books?isbn=8184317379
2.http://www.freebookcentre.net/ComputerScience-Books-Download/Basics-of-Computer-Graphics.html
3. http://www.freetechbooks.com/introduction-to-computer-graphics-t892.html
4. https://en.wikipedia.org/wiki/Book:Game_Devel_Book_2

XIX. Assessment Scheme

Performance indicators		Weightage
Process related(10 Marks)		**30%**
1.	Debugging ability	20%
2.	Follow ethical practices.	10%
Product related (15 Marks)		**70%**
3.	Correctness of algorithm	15%
4.	Correctness of Program codes	30%
5.	Quality of input/output messaging and output formatting	10%
6.	Timely Submission of report	5%
7.	Answer to sample questions	10%
Total (25 Marks)		**100%**

List of Students /Team Members

1. ……..
2. ……..
3. ……..
4. ……

			Dated signature of Teacher
Process Related(10)	**Product Related(15)**	**Total(25)**	

First Semester:

1	Fundamentals of ICT	22001
2	English	22101
3	English Work Book	22101
4	Basic Science (Chemistry)	22102
5	Basic Science (Physics)	22102

Second Semester:

1	Bussiness Communication Using Computers	22009
2	Computer Peripherals & Hardware Maintenace	22013
3	Web Page Design with HTML	22014
4	Applied Science (Chemistry)	22202
5	Applied Science (Physics)	22202
6	Applied Machines	22203
7	Basic Surveying	22205
8	Applied Science (Chemistry)	22211
9	Applied Science (Physics)	22211
10	Fundamental of Electrical Engineering	22212
11	Elements of Electronics	22213
12	Elements of Electrical Engineering	22215
13	Basic Electronics	22216
14	'C' programming Language	22218
15	Basic Electronics	22225
16	Programming in "C"	22226
17	Fundamentals of Chemical Engineering	22231

Third Semester:

1	Applied Multimedia Techniques	22024
2	Advanced Serveying	22301
3	Highway Engineering	22302
4	Mechanics of Structures	22303
5	Building Construction	22304
6	Concrete Technology	22305
7	Strength Of Materials	22306
8	Automobile Engines	22308
9	Automobile Transmission System	22309
10	Mechanical Operations	22313
11	Technology Of Inorganic Chemicals	22314
12	Object Oriented Programming Using C++	22316
13	Data Structure Using 'C'	22317
14	Computer Graphics	22318
15	Database Management System	22319
16	Digital Techniques	22320
17	Principles Of Database	22321
18	Digital Techniques & Microprocessor	22323
19	Electrical Circuits	22324
20	Electrical & Electronic Measurment	22325
21	Fundamental Of Power Electronics	22326
22	Electrical Materials & Wiring Practice	22328
23	Applied Electronics	22329
24	Electrical Circuits & Networks	22330
25	Electronic Measurments & Instrumentation	22333
26	Principles Of Electronics Communication	22334
27	Thermal Engineering	22337
28	Engineering Matrology	22342
29	Mechanical Engineering Materials	22343
30	Theory Of Machines	22344

Fourth Semester:

1	Hydraulics	22401
2	Geo Technical Engineering	22404
3	Chemical Process Instrumentation & Control	22407
4	Fluid Flow Operation	22409
5	Technology Of Organic Chemicals	22410
6	Java Programming	22412
7	GUI Application Development Using VB.net	22034
8	Microprocessor	22415
9	Database Managment	22416
10	Electric Motors And Transformers	22418
11	Industrial Measurements	22420
12	Digital Electronics And Microcontroller Applications	22421
13	Linear Integrated Circuits	22423
14	Microcontroller & Applications	22426
15	Basic Power Electronics	22427
16	Digital Communication Systems	22428
17	Mechanical Engineering Measurments	22443
18	Fluid Mechanics and Machinery	22445
19	Fundamentals Of Mechatronics	22048

Fifth Semester:

1	Design of Steel and RCC Structures	22502
2	Public Health Engineering	22504
3	Heat Transfer Operation	22510
4	Environmental Technology	22511
5	Operating Systems	22516
6	Advanced Java Programming	22517
7	Software Testing	22518
8	Control Systems and PLC's	22531
9	Embedded Systems	22532
10	Mobile and Wireless Communication	22533
11	Industrial Machines	22523
12	Switchgear and Protection	22524
13	Energy Conservation and Audit	22525
14	Power Engineering and Refrigeration	22562
15	Solid Modeling and Additive Manufacturing	22053
16	Guidelines & Assessment Manual for Micro Projects & Industrial Training	22057

Sixth Semester:

1	Solid Modeling	17063
2		17602
3	Contracts & Accounts	17603
4		17604
5	Industrial Fluid Power	17608
6		17610
7		17617
8		17618
9	Software Testing	17624
10		17625
11	Mobile Computing	17632
12	System Programing	17634
13		17637
14	Power Electronics	17638
15		17639
16		17643
17		17646
18		17648
19		17656
20		17657
21	Embedded System	17658
22		17663
23		17664
24	Industrial Drives	17667
25	Video Engineering	17668
26		17669
27		17671
28		17672
29		17673

First Year:

1	Pharmaceutics - I	0805
2		0806
3	Pharmacognosy	0807
4		0808
5		0809

Second Year:

1	Pharmaceutics - II	0811
2		0812
3		0813
4		0816

HEAD OFFICE

Secretary,

Maharashtra State Board of Technical Education

49, Kherwadi, Bandra (East), Mumbai - 400 051

Maharashtra (INDIA)

Tel: (022)26471255 (5 -lines)

Fax: 022 - 26473980

Email: -secretary@msbte.com

Web -www.msbte.org.in

REGIONAL OFFICES:

MUMBAI

Deputy Secretary (T),

Mumbai Sub-region,

2nd Floor, Govt. Polytechnic Building,

49, Kherwadi, Bandra (East)

Mumbai - 400 051

Phone: 022-26473253 / 54

Fax: 022-26478795

Email: rbtemumbai@msbte.com

PUNE

Deputy Secretary (T),

M.S. Board of Technical Education,

Regional Office,

412-E, Bahirat Patil Chowk,

Shivaji Nagar, Pune

Phone: 020-25656994 / 25660319

Fax: 020-25656994

Email: rbtepn@msbte.com

NAGPUR

Deputy Secretary (T),

M.S. Board of Technical Education

Regional Office,

Mangalwari Bazar, Sadar, Nagpur - 440 001

Phone: 0712-2564836 / 2562223

Fax: 0712-2560350

Email: rbteng@msbte.com

AURANGABAD

Deputy Secretary (T),

M.S. Board of Technical Education,

Regional Office,

Osmanpura, Aurangabad -431 001.

Phone: 0240-2334025 / 2331273

Fax: 0240-2349669

Email: rbteau@msbte.com