**Geometrical Drawing**

**A mini project report submitted in partial fulfilment of the requirement for the Award of the Degree of**

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

***by***

**B.Sravan Kumar (160717733025)**

**T.Sai Charan Reddy (160717733008)**

**V.Jeshwanth (160717733014)**

***Under the Guidance of***

**Mr. K. Kishore Kumar, Assistant Professor, CSE**

****

**Department of Computer Science and Engineering**

**Methodist College of Engineering and Technology**

**King Koti, Abids, Hyderabad-500001.**

**2018-2019**

**Methodist College of Engineering and Technology,**

**King Koti, Abids, Hyderabad-500001,**

**Department of Computer Science and Engineering**



**DECLARATION BY THE CANDIDATES**

We, B. **Sravan Kumar (160717733025), T. Sai Charan Reddy (160717733008) and V. Jeshwanth (160717733014)** students of Methodist College of Engineering and Technology, pursuing Bachelor’s degree in Computer Science and Engineering, here by declare that this mini project report entitled “**Geometrical Drawing",** carried out under the guidance of **Mr. K. Kishore Kumar** submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science. This is a record work carried out by us and the results embodied in this project have not been reproduced/copied from any source.

**B. Sravan kumar (160717733025)**

**T. Sai Charan Reddy (160717733008)**

**V. Jeshwanth(160717733014)**

**Methodist College of Engineering and Technology,**

**King Koti, Abids, Hyderabad-500001.**

**Department of Computer Science and Engineering**



**CERTIFICATE BY THE SUPERVISOR**

This is to certify that this project report entitled “ Geometrical Drawing ” being submitted by B. Sravan Kumar (160717733025),T. Sai Charan Reddy(160717733008) and V. Jeshwanth (160717733014), submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering, during the academic year 2018-2019, is a bonafide record of work carried out by them.

**Mr. K. Kishore Kumar**

Assistant Professor

Date:

**Methodist College of Engineering and Technology,**

**King Koti , Abids , Hyderabad-500001.**

**Department of Computer Science and Engineering**



**CERTIFICATE BY THE HEAD OF THE DEPARTMENT**

This is to certify that this project report entitled “**Geometrical Drawing**” *by* **B. Sravan Kumar** (160717733025), **T. Sai Charan Reddy** (160717733008) **and V. Jeshwanth** (160717733014),submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering of the Osmania University, Hyderabad, during the academic year 2018-19, is a bonafide record of work carried out by them.

**Mrs. P. Lavanya**

Associate Professor &

Head of the Department.

.

**DATE:**

**ACKNOWLEDGEMNT**

We would like to express our sincere gratitude to our project guide **Mr. K. Kishore Kumar , Assistant Professor,** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without his innovative ideas and his relentless support and encouragement.

We would like to thank our project coordinator **Mr. Sandeep R, Assistant Professor**  he helped us by being an example of high vision and pushing towards greater limits of achievement.

Our sincere thanks to **Mrs. P. Lavanya, Associate Professor** and **Head of the Department of Computer Science and Engineering,** for her valuable guidance and encouragement which has played a major role in the completion of the project and for helping us by being an example of high vision and pushing towards greater limits of achievement.

We would like to express a deep sense of gratitude towards the **Dr. G. Ravinder Reddy, Principal, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We would like to express a deep sense of gratitude towards the **Dr. Lakshmipathi Rao, Director, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We are indebted to the Department of Computer Science & Engineering and Methodist College of Engineering and Technology for providing us with all the required facility to carry our work in a congenial environment. We extend our gratitude to the CSE Department staff for providing us to the needful time to time whenever requested.

We would like to thank our parents for allowing us to realize our potential, all the support they have provided us over the years was the greatest gift anyone has ever given us and also for teaching us the value of hard work and education. Our parents has offered us with tremendous support and encouragement, thanks to our parents for all the moral support and the amazing opportunities they have given us over the years.

***ABSTRACT***

The project titled “Geometrical Drawing” is used to design and calculate the area and structure of different shapes like circle, rectangle, conic etc... By this project the user gets the task easier. For example if the user needs to design and visualize a cone and find the area of that. By this program the work is getting easier. So we come up with this idea “Geometrical Drawing” in which the user visualizes the diagram and find the area of that. Users just need to enter the measurements and then user get the design as per given measurements, and along with its area. We are using c++ language and graphics concept to develop this program.

i

**Contents:**

**Abstract i**

**Contents ii**

**List of figures iv**

**List of table vii**

**1. Introduction..……………………………………………………………………….........**.**01**

2**. Literature survey……….……………………………………………………………….02**

**3. Design analysis…………………………….…………………………………………….03**

3.1 Existing System…………………………………………………………….…….03

3.2 Proposed System…………………………………….……………………….…….03

3.3 Block diagram……………………………………………………………….…….04

3.4 Architecture diagram………………………………………………………….…..05

3.5 Main flow chart…….………………………………………………………….…..06

**4. Graphical user interface…………………………………………………….……….....08**

4.1 Input…………………………………………………………………………….08

4.2 Output……………………………………………….………………...………..08

**5.Implementation……………………………………………………………………….…09**

5.1 Flowchart………………………………………………………..……………….….09

5.2 Module……………………………………………………….………….……………..11

**6. Testing……………………………………………………………………...……………12**

6.1. Individual Module testing…………………..…………………………………..12

6.1.1.Main menu…………………………………………………….………12

6.1.2.Circle……………….…….……………………………….……..……..13

6.1.3.Ellipse………………….…………………………………..……….….13

6.1.4.Line…….……………………………………………………………….14

6.1.5.Cone…………………………………………………………………...14

6.1.6.Arc……………………………………………………………..……....15

6.1.7.Rectangle………………………………………………………………..15

ii

6.2.Screenshots………………………………………………………………….….16

6.2.1. Circle……………………………………………………….......……16

6.2.2. Ellipse………………………………………………………..………18

6.2.3. Line……….…………………………………………….….……..….20

6.2.4. Cone………..…………………………………………….…...…..…22

6.2.5. Arc…………………………………………………………….……..24

6.2.6. Rectangle……………………………………………………………..26

**Conclusion……………………….…………….……….……..…………………………...29**

**References……………………………………….……….……..………………………….30**

**Appendix A:Used header files………………………….………………………………….31**

**Appendix B: Sample code………………………………….……………………………..32**

**Appendix C: Software and hardware requirements……………..………………………36**

**Appendix D: Technologies used…………………………….……………………………37**

Iii

**List of figures:**

**Figure: 3.3.1 Block diagram**

**Figure: 3.4.1 Architecture diagram**

**Figure: 3.5.1 Flow chart**

**Figure: 4.1.1 Number keys**

**Figure: 5.1 Flow chart**

**Figure 6.2.1 Intro**

**Figure 6.2.1.1 Main menu circle**

**Figure 6.2.1.2 circle inputs**

**Figure 6.2.1.3 calculations of circle**

**Figure 6.2.1.4 circle design**

**Figure 6.2.2.1 Main menu Ellipse**

**Figure 6.2.2.2 Ellipse inputs**

**Figure 6.2.2.3 calculations of ellipse**

**Figure 6.2.2.4 Ellipse design**

**Figure 6.2.3.1 Main menu Line**

**Figure 6.2.3.2 Line inputs**

**Figure 6.2.3.3 calculations of line**

**Figure 6.2.3.4 line design**

**Figure 6.2.4.1 Main menu Cone**

**Figure 6.2.4.2 cone inputs**

**Figure 6.2.4.3 calculations of cone**

**Figure 6.2.4.3 cone design**

**Figure 6.2.5.1 Main menu Arc**

**Figure 6.2.5.2 Arc inputs**

**Figure 6.2.5.3 calculations of arc**

**Figure 6.2.5.4 Arc design**

**Figure 6.2.6.1 Main menu Rectangle**

iv

**Figure 6.2.6.2 Rectangle inputs**

**Figure 6.2.6.3 calculations of rectangle**

**Figure 6.2.6.4 Rectangle design**

v

**List of tables**

**Table 6.1.1: Testing on the main menu**

**Table 6.1.2: Testing on circle**

**Table 6.1.3: Testing on ellipse**

**Table 6.1.4: Testing on line**

**Table 6.1.5: Testing on cone**

**Table 6.1.6: Testing on arc**

**Table 6.1.7: Testing on rectangle**

vi

**1. INTRODUCTION:-**

The project “Geometrical drawing", which is developed is to provide the flexible solution for the user.This project will teach you to create a program in C++ to Calculate the area and structure of different shapes like circle, rectangle, conic etc.. Ex. If the user needs to design and visualize a cone and find the area of that. By this program the work is getting easier. User just needs to enter the measurements and then user can get the design as per the given measurements, and along with its area. This will be so helpful for the user to visualize the geometric shape by giving the required parameters.

So in this project 6 basic geometrical shapes were designed. They are circle, ellipse, line, cone, arc, and rectangle.

1

**2. LITERATURE SURVEY:**

* There are many scenarios in which some are:

-use to draw and visualize with the hand and calculate

-so much time consuming

-makes the task complicated

* Our main aim is to create this geometrical drawing which eliminates the disadvantages.

In our project:

-makes the user task easier.

-by giving required measurements we can get the final design.

- user can visualize the different shapes.

- user can get the calculations of particular shape as per the given parameters.

2

**3. DESIGN ANALYSIS:**

3.1 Existing System:

There are many scenarios in which some are: use to draw with the hand and calculate, so much time consuming, makes the task complicated and many calculation errors.

3.2 Proposed System:

Our main aim is to create this geometrical drawing which eliminates the disadvantages.

In our project:

-makes the user task easily.

-by giving required measurements user can get the final design.

3

3.3 Block diagram:

This is the basic block diagram. Once we execute the program, initially the main menu is displayed on the screen, and then the user has to choose a geometrical shape. The inputs(parameters) are displayed on the screen. Then the user visualizes the design. Once it is completed, the calculation part of the respective shape is displayed.

**Figure: 3.3.1 – block diagram**

4

3.3 Architecture diagram:

The lists of geometrical shapes are circle, ellipse, line, cone, arc, and rectangle.

So once we select a shape inputs (parameters given by user) calculations part then the design is displayed main menu is displayed again on screen and process is continued.

**Figure: 3.3.2 –** Architecture diagram

5

3.4 Main flow chart:

Once we execute the program the main menu is displayed on the screen with Circle, ellipse, line cone, arc, and rectangle. There is a switch case in which you will select a game and get the design. when you enter an invalid number then the message will be displayed that ‘enter a valid option’. Once the game is done, the menu is displayed again. To exit, 5 will be your option.

6

**Figure: 3.4.3 –**Main flow chart

7

**4. GRAPHICAL USER INTERPHASE:**

4.1 Input

In this project, inputs are entered using a keyboard.

Numerical keys used in the main menu, circle, ellipse, line, cone, arc, rectangle.

 Figure 4.1.1 number keys

4.2 Output

The computer generates the design and calculation part.

In circle, ellipse, line, cone, arc, rectangle the computer generates the design and calculation part of the respective shape.

# 8

# 5. IMPLEMENTATION

5.1 Flowchart:

9

**Figure: 5.1.1–**flow chart

10

5.2 Module

Step 1: Menu

Step 2: Displays 7 options.

1. Circle
2. Ellipse
3. Line
4. Cone
5. Arc
6. Rectangle
7. Exit.

Step 3: The user has to choose an option.

Step 4: If it is (i) it takes the input as radius and displays its area , circumference and diameter(calculation part) and shape of the circle

Step 5: If it is (ii) it takes the input as radius of x , y-axis and display its area , length of major , minor axis and eccentricity and then displays shape of the ellipse.  
Step 6: If it is (iii) it takes the input as (x1,y1) for first point and (x2,y2) for second point and displays its slope and distance of line and displays the line.

Step 7: If it is (iv) it takes the input as radius then displays the area of cone and displays the cone shape.

Step 8: If it is (v) it takes the input as angles and radius then displays the arc length and then displays the arc shape.

Step 9: If it is (vi) it takes input as length and breadth then displays the area , perimeter and diagonal and displays the rectangle shape.

Step 10:If it is (vii) then it will exit or the user can continue.

11

**6. TESTING:**

Testing is done to discover errors.

There are two methodologies in testing which include unit and integrated testing.

Unit testing is verifying each module separately. The errors if any are verified.

Integration testing is verifying all modules sequentially in a program and rectifying the errors.

6.1. Individual Module testing

6.1.1 Main menu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test cases** | **Step number** | **Case description** | **Expected results** | **Actual results** | **Pass/ fail** |
| Main menu | Step 1 | Purpose: this verified the proper functioning of the menu |  |  |  |
|  | Step 2 | Enter the numerical 1, 2 3, 4, 5, 6, 7 | Displays the respective shape | Displays the respective shape | Pass |

12

6.1.2 Circle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test cases | Step number | Case description | Expected results | Actual results | Pass/ fail |
| Circle | Step 1 | Purpose: this verifies the proper functioning of the shape |  |  |  |
|  | Step 2 | The input of the numerical  for the radius | Design of the circle | Design of the circle | Pass |
|  | Step 3 | For the given parameters  Calculation part | Area  Circumferene  Diameter | Area  Circumferene  Diameter | Pass |

6.1.3 Ellipse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test cases | Step number | Case description | Expected results | Actual results | Pass/ fail |
| Ellipse | Step 1 | Purpose: this verifies the proper functioning of the shape |  |  |  |
|  | Step 2 | The input of the numerical  for the radius of x-axis and y-axis | Design of the  Ellipse | Design of the Ellipse | Pass |
|  | Step 3 | For the given parameters  Calculation part | Area  Length of axis  Eccentricity | Area  Length of axis  Eccentricity | Pass |

13

6.1.4 Line

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test cases | Step number | Case description | Expected results | Actual results | Pass/ fail |
| Line | Step 1 | Purpose: this verifies the proper functioning of the shape |  |  |  |
|  | Step 2 | The input of the numerical  for the points | Design of the Line | Design of the Line | Pass |
|  | Step 3 | For the given parameters  Calculation part | Slope and  Distance of the line | Slope and  Distance of the line | Pass |

6.1.5 Cone

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test cases | Step number | Case description | Expected results | Actual results | Pass/ fail |
| Cone | Step 1 | Purpose: this verifies the proper functioning of the shape |  |  |  |
|  | Step 2 | The input of the numerical  for the radius | Design of the cone | Design of the cone | Pass |
|  | Step 3 | For the given parameters  Calculation part | Area of the  cone | Area of the cone | Pass |

14

6.1.6 Arc

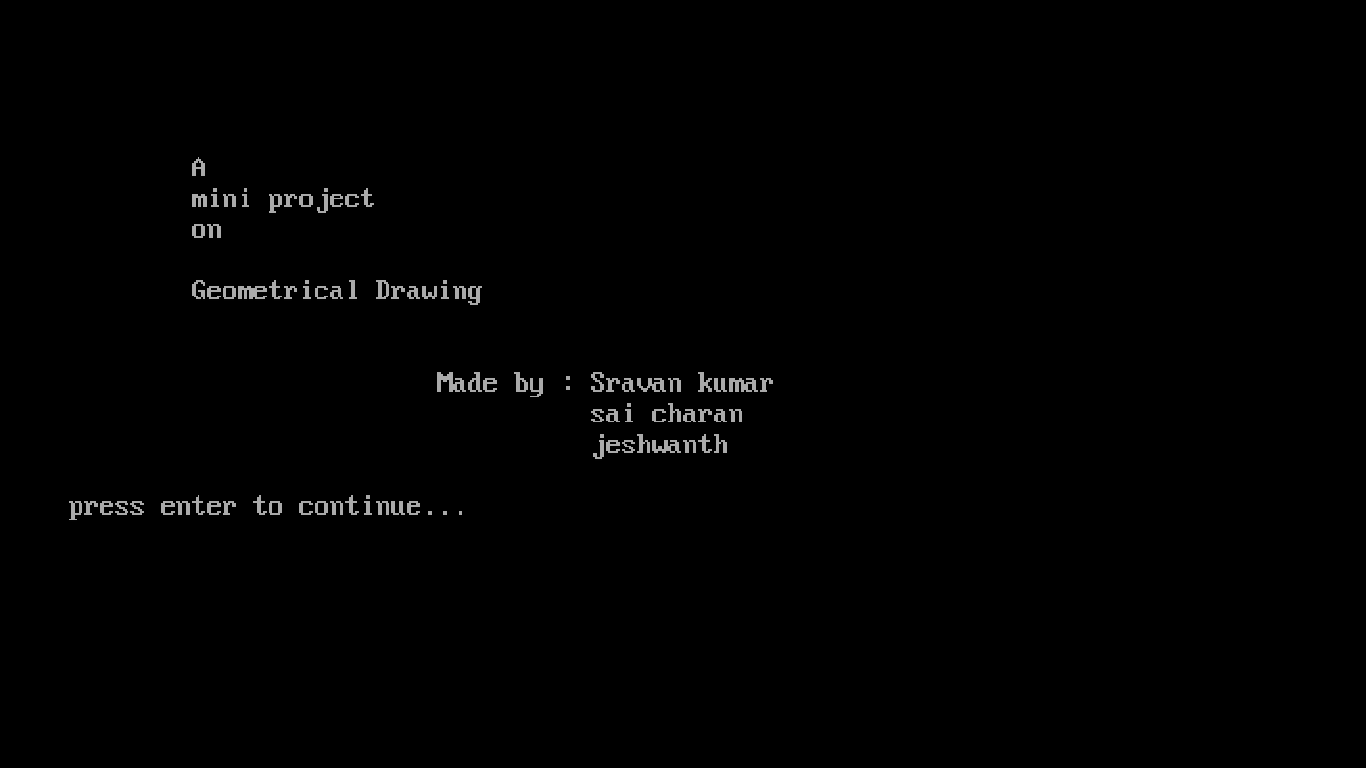
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test cases | Step number | Case description | Expected results | Actual results | Pass/ fail |
| Arc | Step 1 | Purpose: this verifies the proper functioning of the shape |  |  |  |
|  | Step 2 | The input of the numerical  for the angles and radius | Design of the Arc | Design of the Arc | Pass |
|  | Step 3 | For the given parameters  Calculation part | Length of Arc | Length of Arc | Pass |

6.1.7 Rectangle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test cases | Step number | Case description | Expected results | Actual results | Pass/ fail |
| Rectangle | Step 1 | Purpose: this verifies the proper functioning of the shape |  |  |  |
|  | Step 2 | The input of the numerical  for the length and breadth | Design of the Rectangle | Design of the Rectangle | Pass |
|  | Step 3 | For the given parameters  Calculation part | Area  Perimeter  Diagonal of  rectangle | Area  Perimeter  Diagonal of  rectangle | Pass |

15

6.2 Screenshots:



**Figure: 6.2.1: Intro part**

6.2.1 Circle

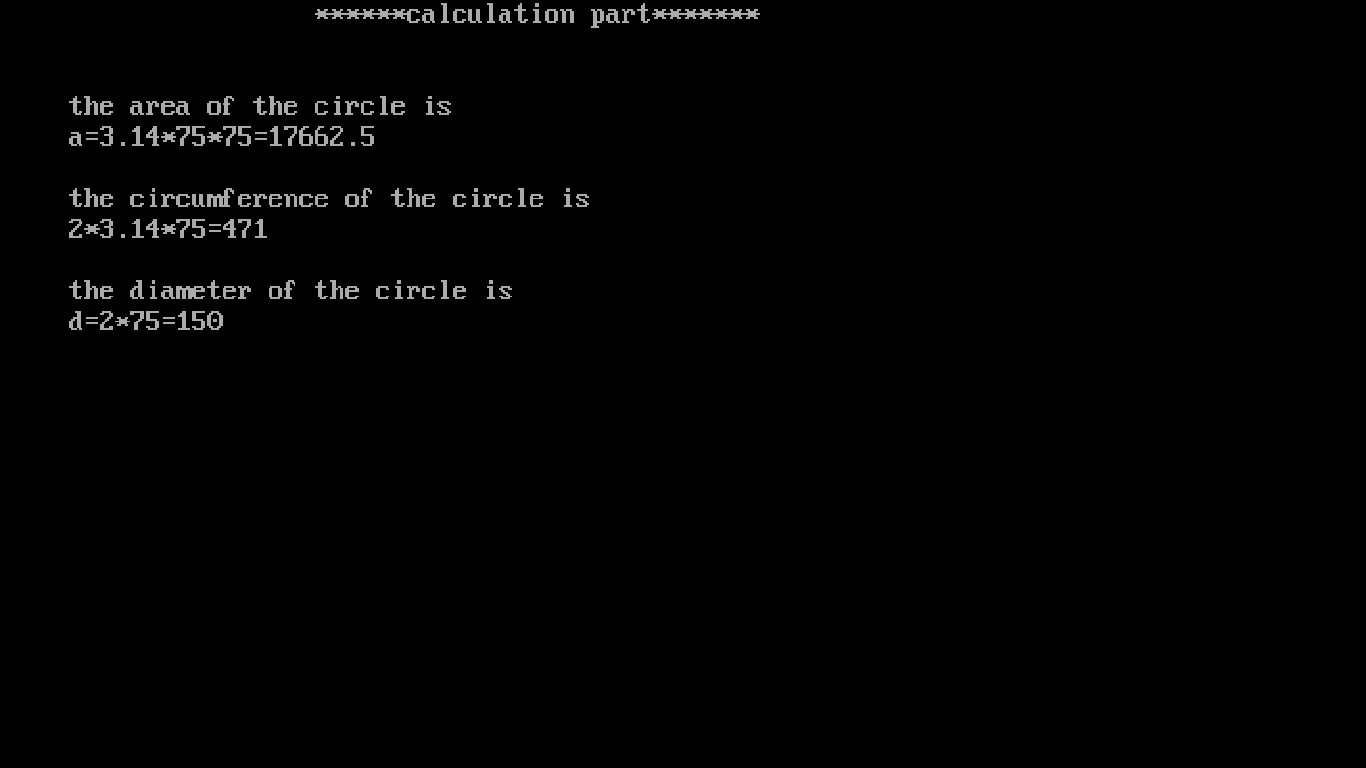


**Figure: 6.2.1.1: main menu Circle selected**

16

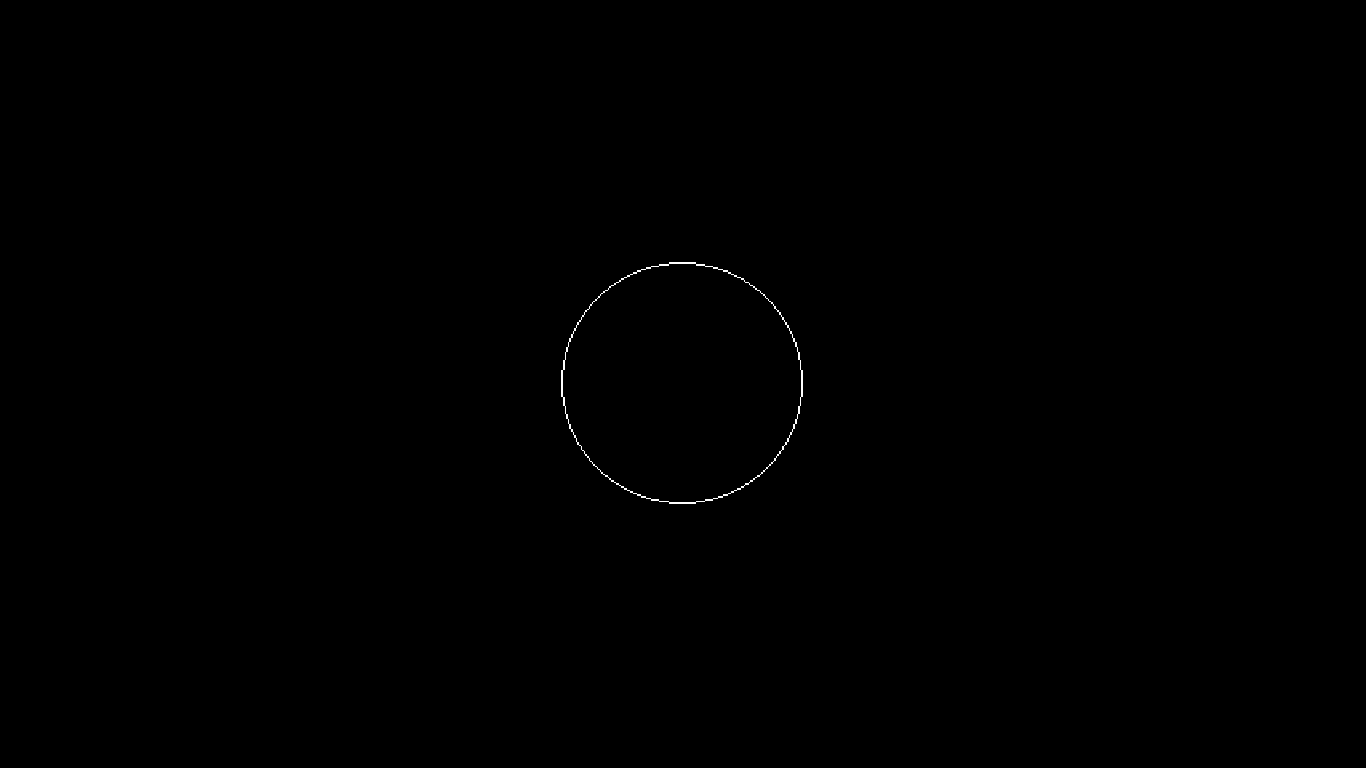


**Figure: 6.2.1.2: circle inputs**



**Figure: 6.2.1.3: calculations of circle**

17



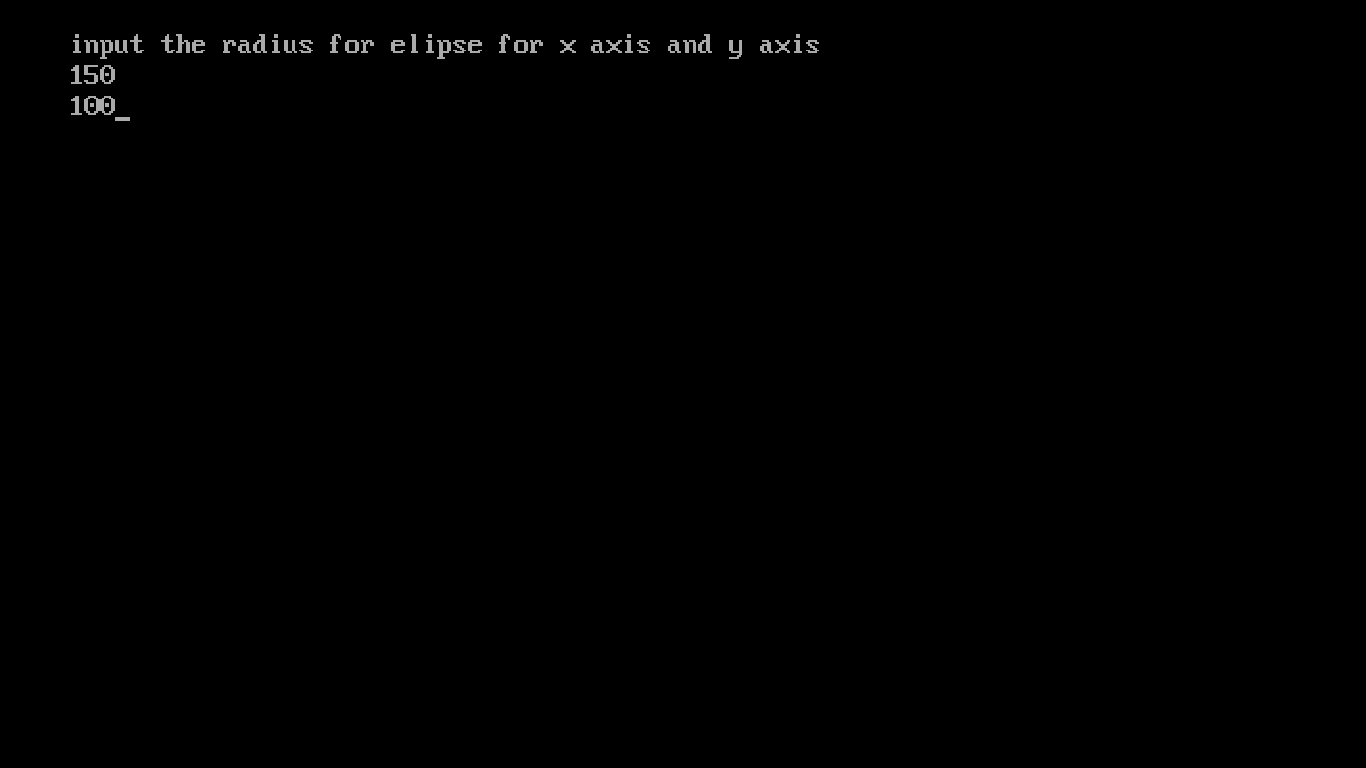
**Figure: 6.2.1.4:circle design**

6.2.2 Ellipse

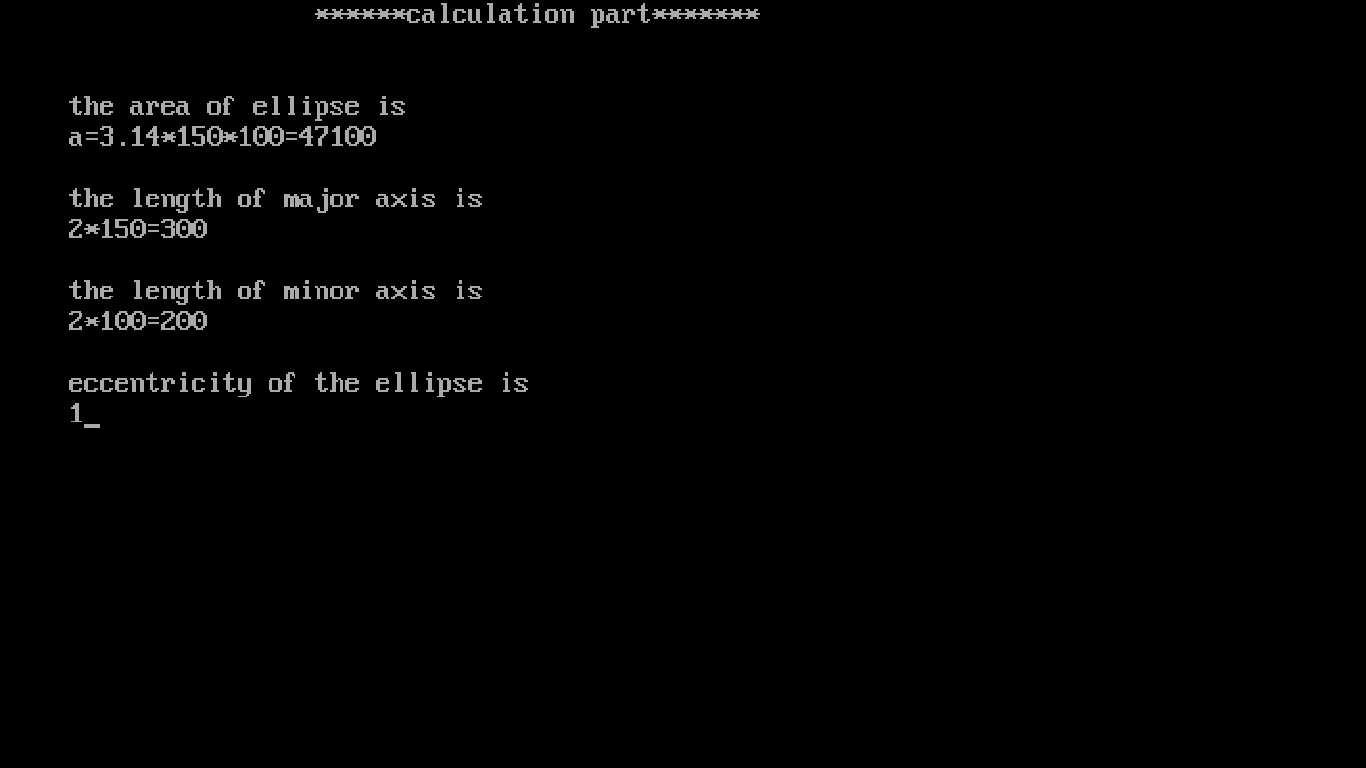


**Figure: 6.2.2.1 main menu ellipse selected**

18

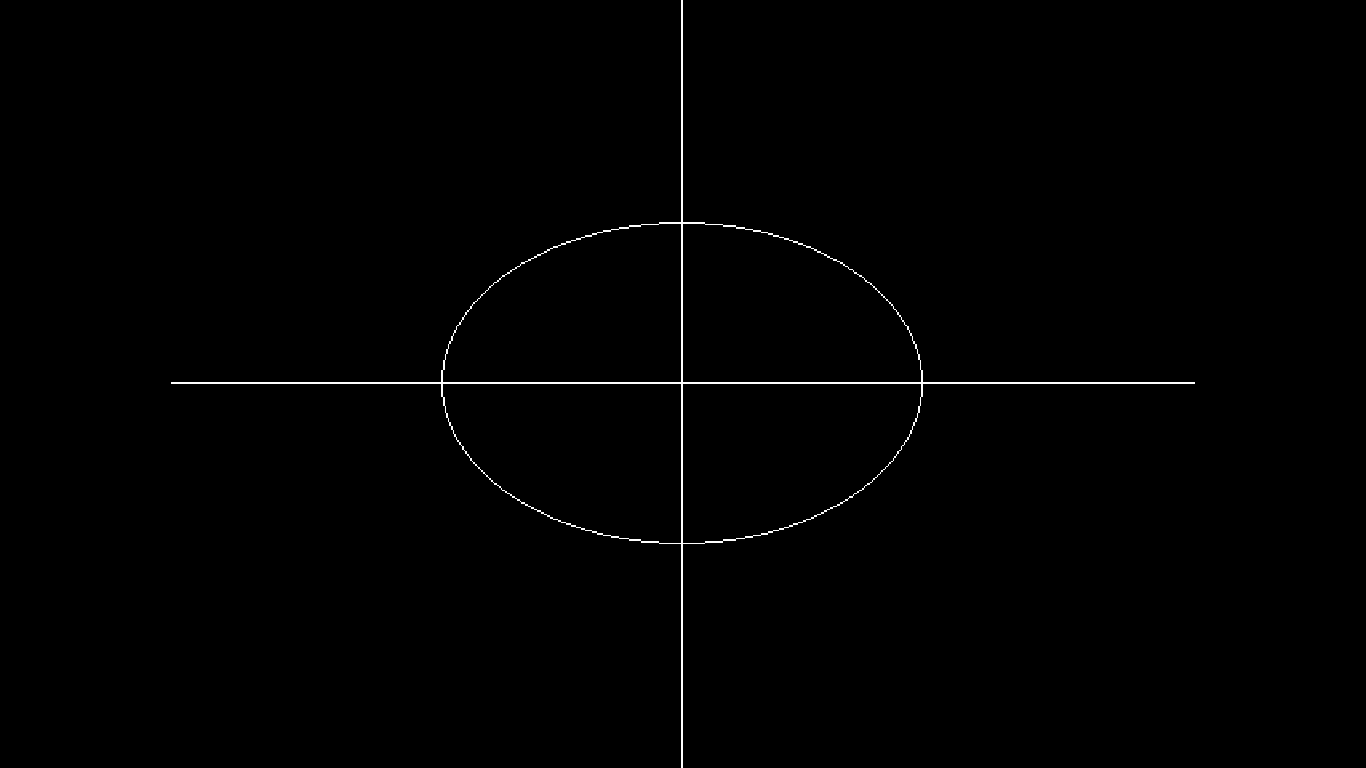


**Figure: 6.2.2.2: ellipse inputs**



**Figure: 6.2.2.3: calculations of ellipse**

19



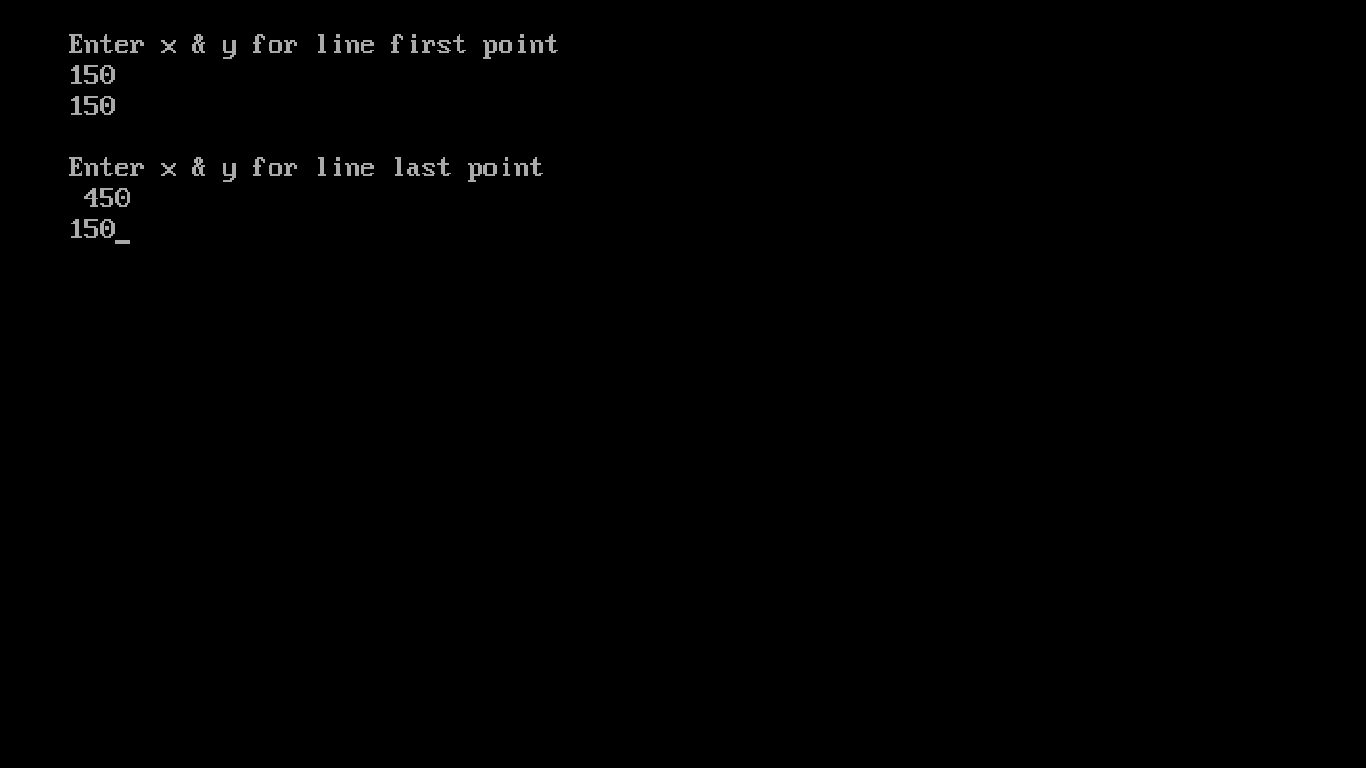
**Figure: 6.2.2.4 ellipse design**

6.2.3 Line



**Figure: 6.2.3.1 main menu line selected**

20

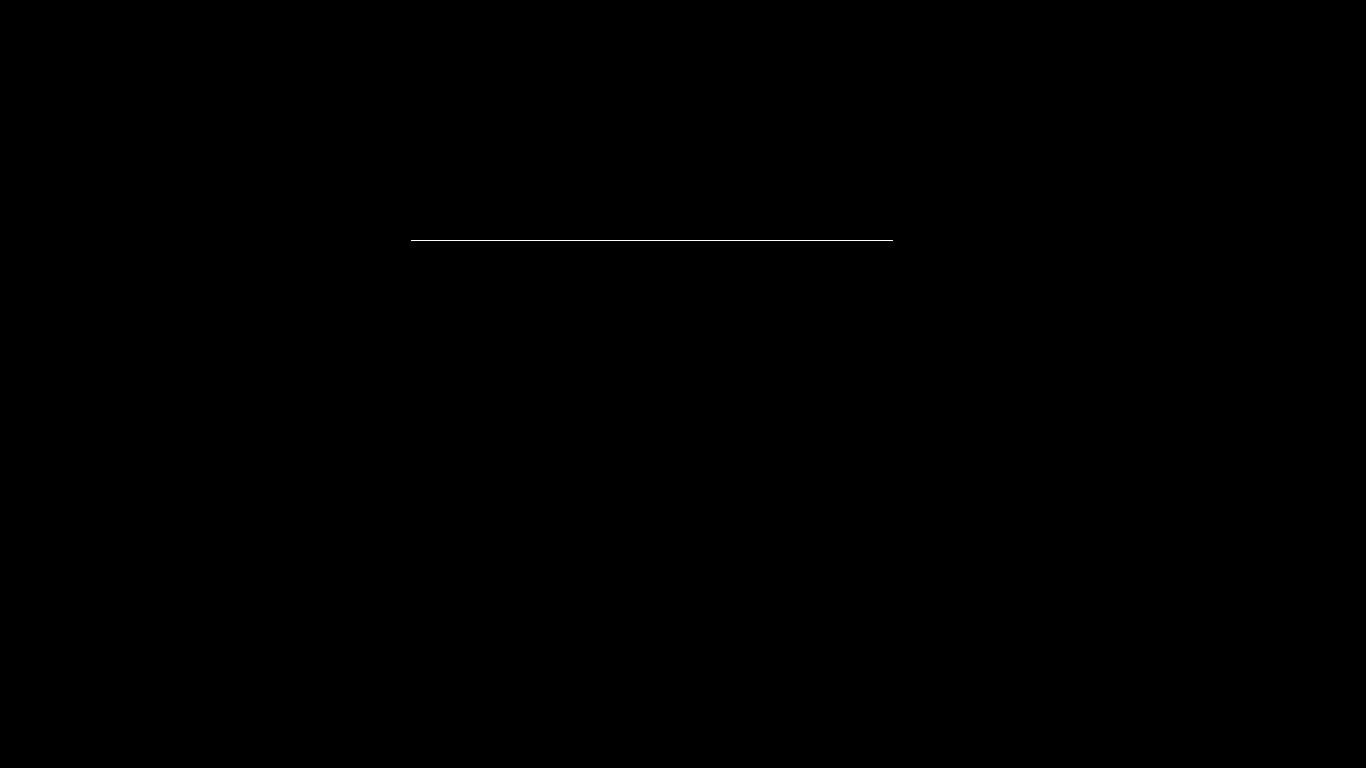


**Figure: 6.2.3.2: line inputs**



**Figure: 6.2.3.3 calculations of line**

21



**Figure: 6.2.3.4 line design**

6.2.4 Cone

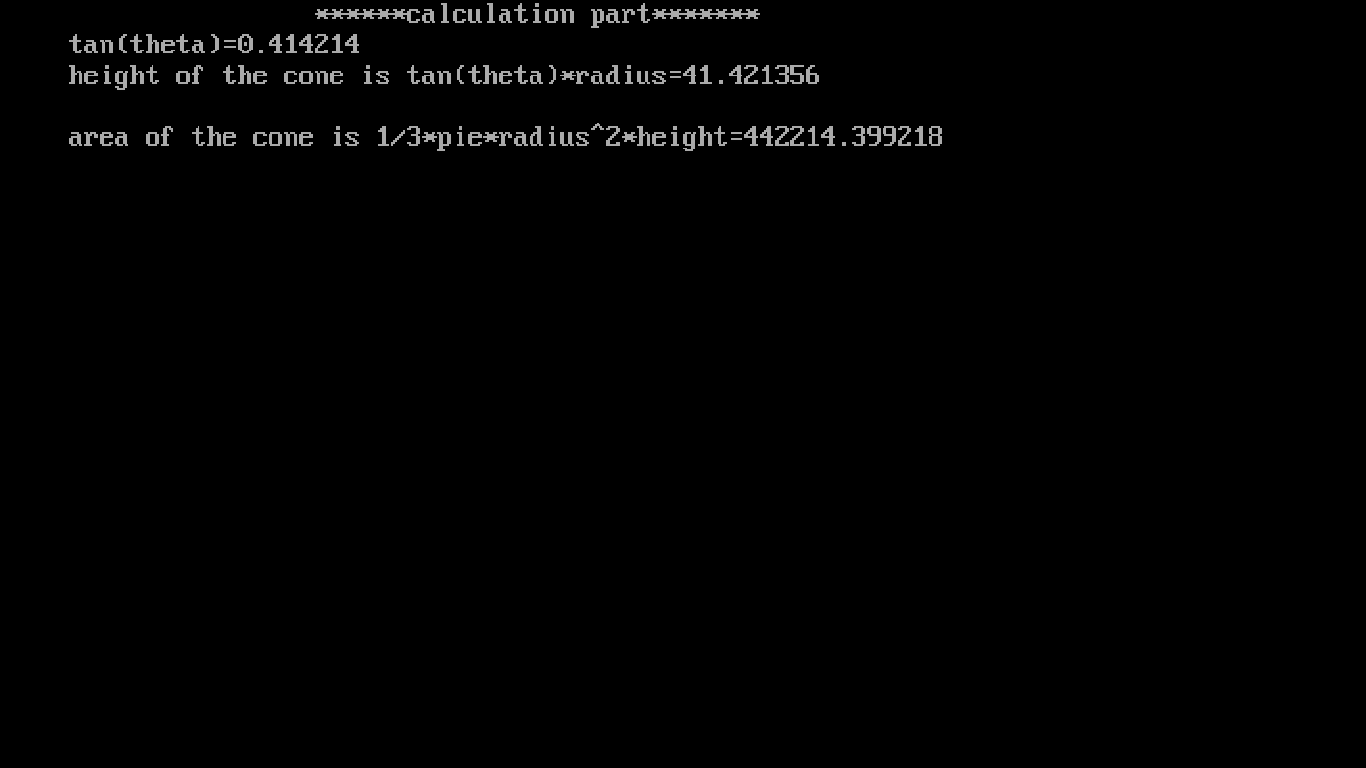


**Figure: 6.2.4.1: main menu cone selected**

22

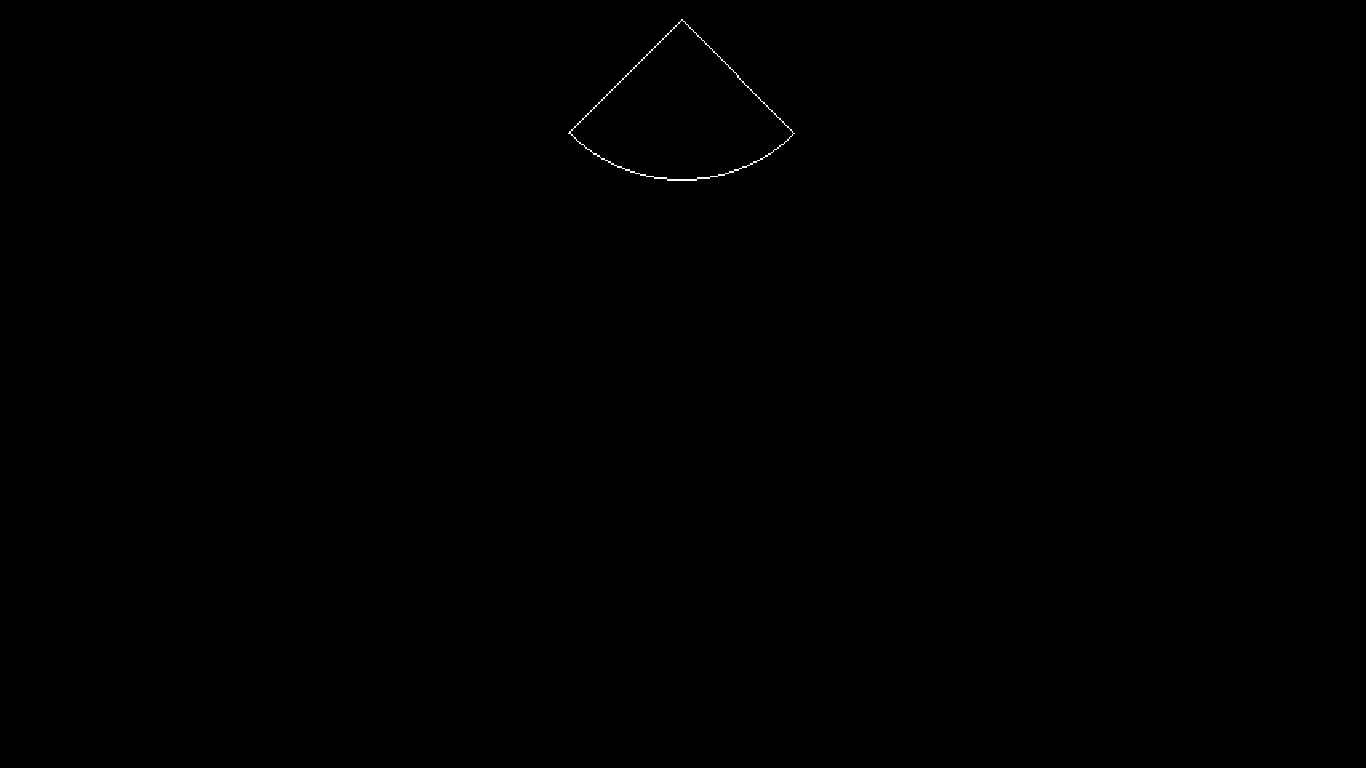


**Figure: 6.2.4.2: cone inputs**



**Figure: 6.2.4.3: calculations of cone**

23



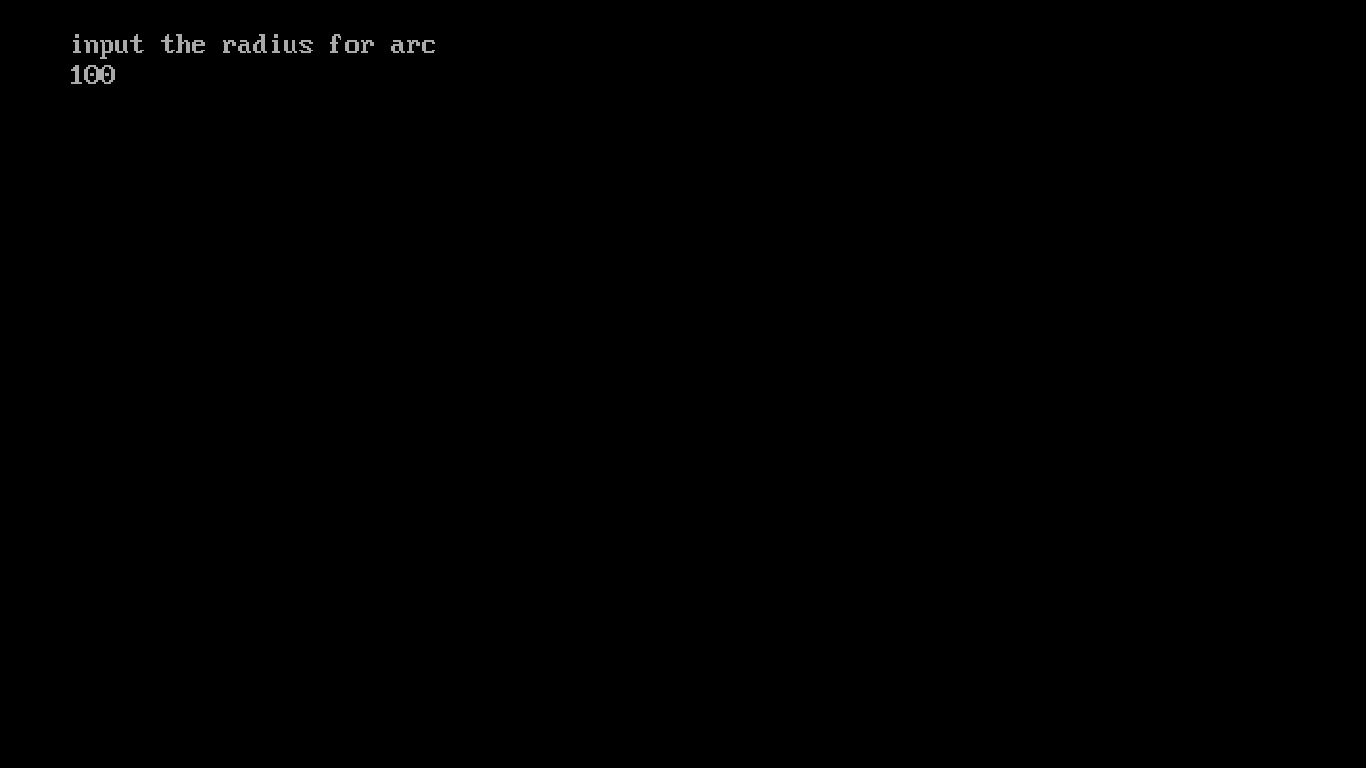
**Figure: 6.2.4.4: cone design**

6.2.5 Arc



**Figure: 6.2.5.1:main menu arc selected**

24

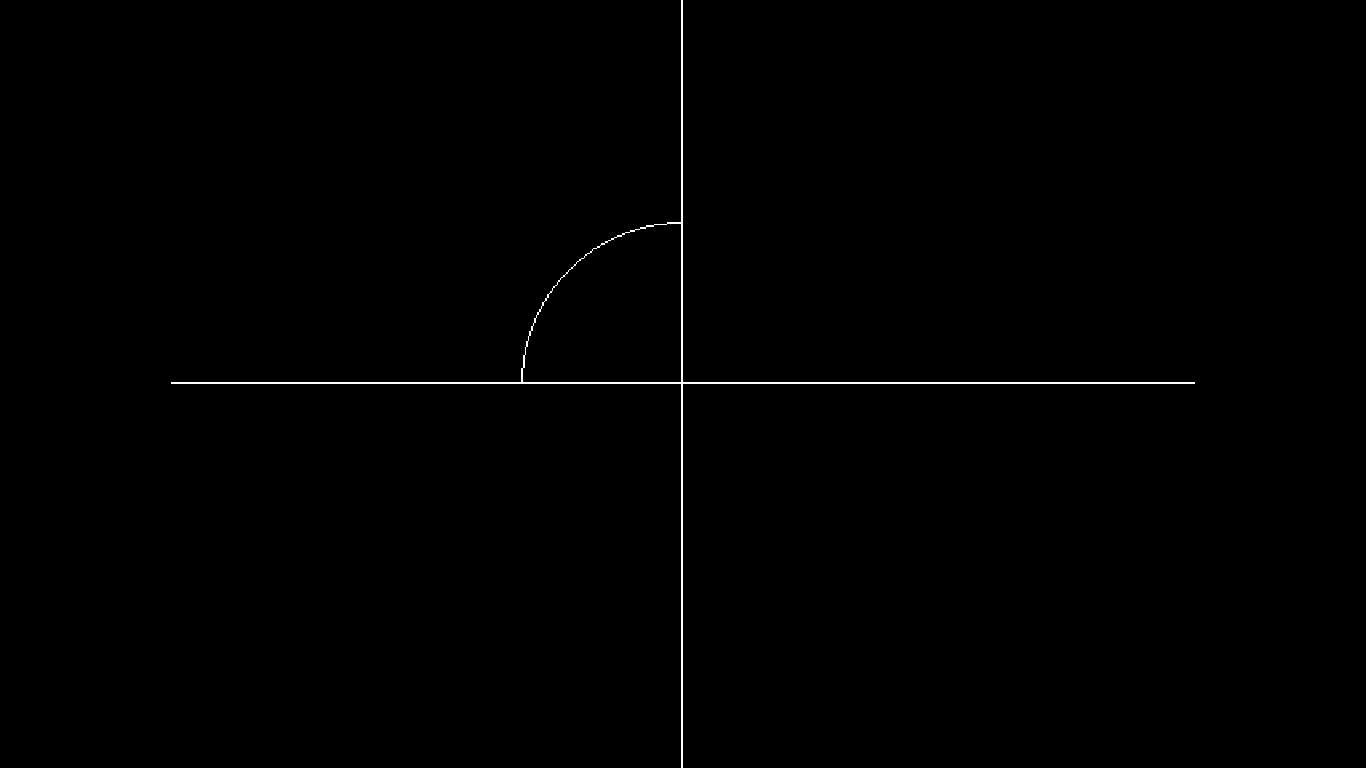


**Figure: 6.2.5.2:arc inputs**



**Figure: 6.2.5.3: calculations of arc**

25

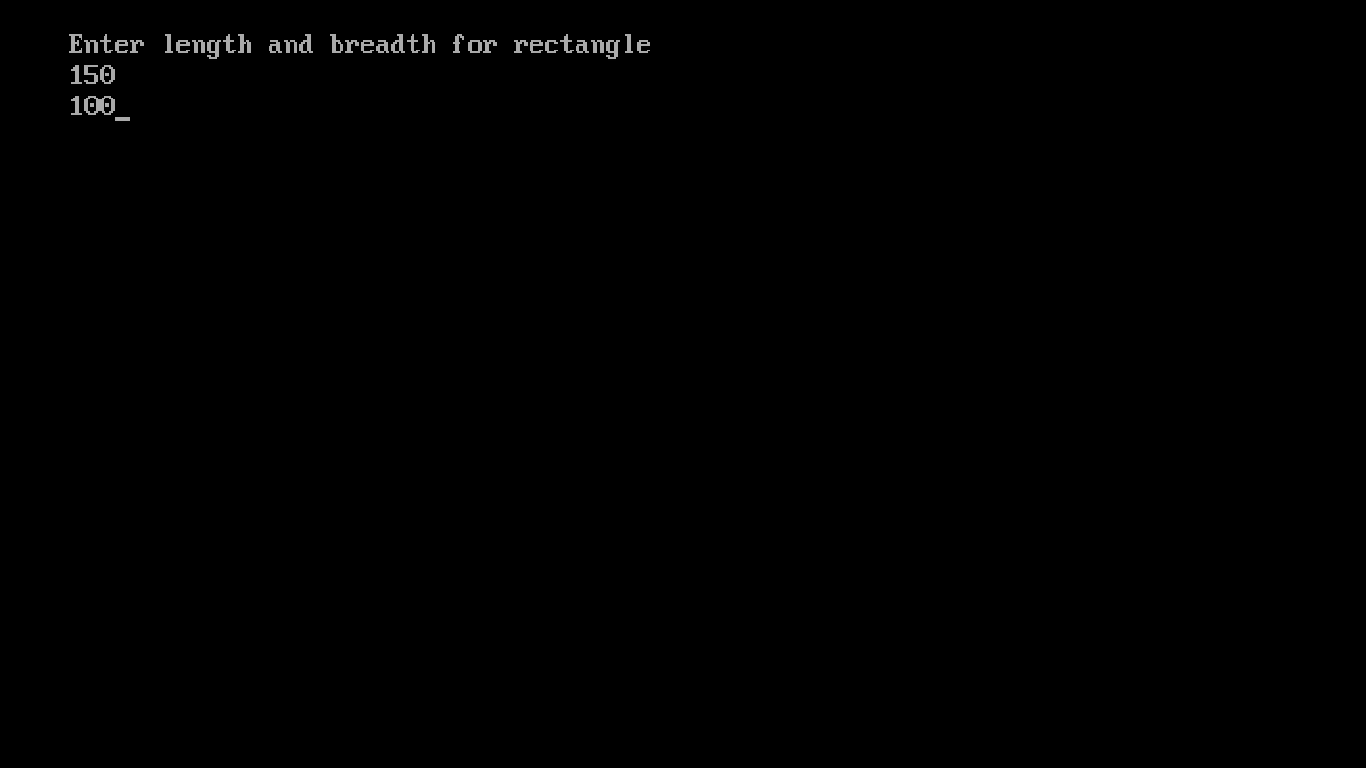
 **Figure: 6.2.5.4: arc design**

6.2.6 Rectangle



**Figure: 6.2.6.1:main menu rectangle selected**

26

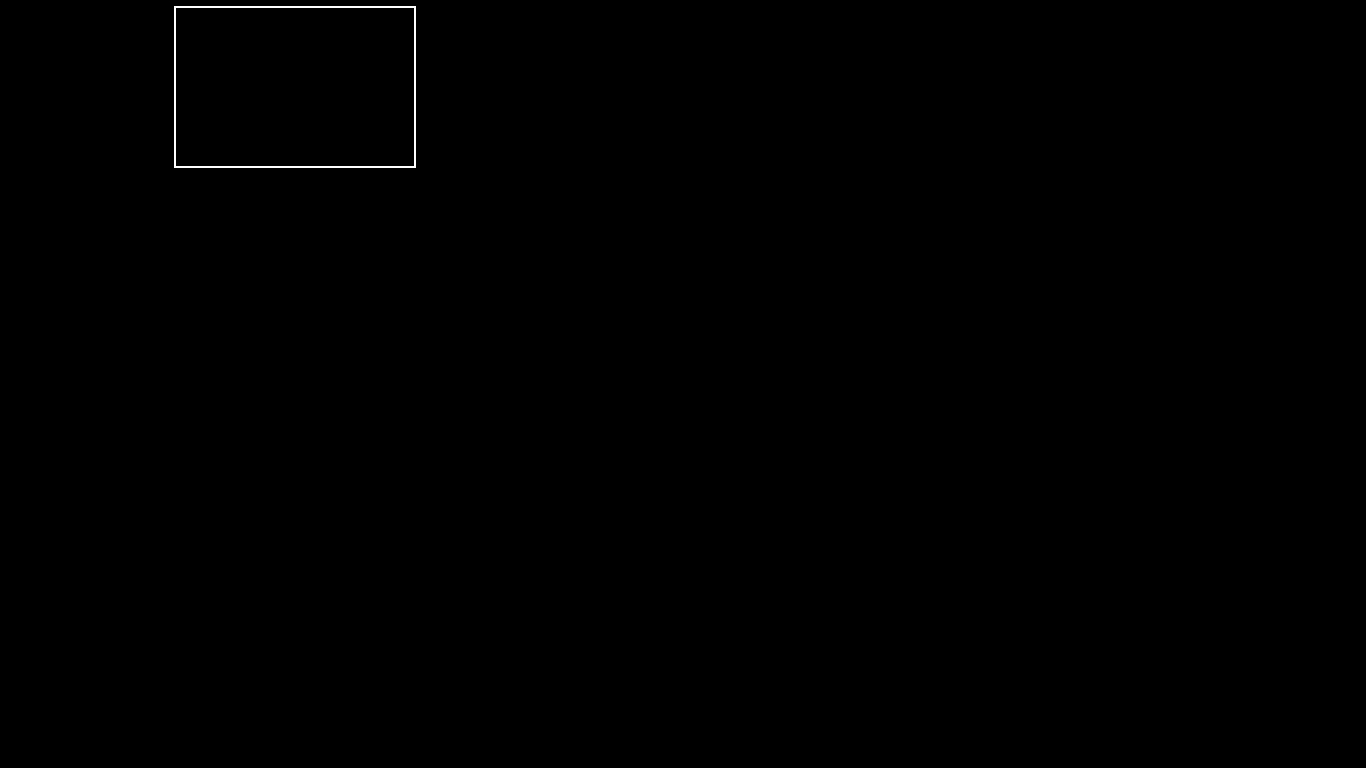


**Figure: 6.2.6.2:rectangle inputs**



**Figure: 6.2.6.3:calculations of rectangle**

27



**Figure: 6.2.6.4: rectangle design**

28

**CONCLUSION:**

This application is designed to make the user task easier. User can get the design as per the given requirements. This program will help you calculate the areas of different structures. In future we can design and add more structures using advanced graphics.

**FUTURE SCOPE**:

In future we can design and add more geometrical shapes using advanced graphics. So that the user can have clear idea on geometrical shapes

29

**REFERENCES:**

1. https://cppprojectcode.blogspot.com/2010/09/geometricaldrawing.html?m=1
2. Object Oriented Programming using C++, Algorithms and Applications in C++;
3. Object oriented programming with c++ by Robert Lafore .

30

**APPENDIX A:**

**USED HEADER FILES**

iostream.h (for cin & cout)

stdlib.h (for exit)

dos.h (for delay )

conio.h (for getche )

cin>>x; (for input)

cout<<“hiii”; (for output)

cin.get() (press enter to continue)

exit(0); (for exiting while loop at menu)

delay(1000); (pause for 1000 milli seconds)

31

**APPENDIX B:**

**SAMPLE CODE**

Each shape has its own instruction.

Circle:-

void circ::cir()

{

int gdriver = DETECT, gmode, errorcode;

int midx, midy;

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

midx = getmaxx()/2;

midy = getmaxy()/2;

setcolor(getmaxcolor());

circle(midx, midy, rad);

getch();

closegraph();

}

Ellipse:-

void elli::para()

{

int gdriver = DETECT, gmode, errorcode;

int midx, midy;

int stangle = 0, endangle = 360;

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:")

getch();

exit(1);}

32

line(getmaxx()/2,0,getmaxx()/2,getmaxy());

line(0,getmaxy()/2,getmaxx(),getmaxy()/2);

midx = getmaxx() / 2;

midy = getmaxy() / 2;

setcolor(getmaxcolor());

line(getmaxx()/2,0,getmaxx()/2,getmaxy());

line(0,getmaxy()/2,getmaxx(),getmaxy()/2);

ellipse(midx, midy, stangle, endangle,

radx, rady);

getch();

closegraph();

}

Line:-

void linee::cline()

{

int gdriver = DETECT, gmode, errorcode;

int maxx, maxy;

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

line(x,y,radx,rady );

getch();

closegraph();}

Cone:-

void conic::coni(){

int gdriver = DETECT, gmode, errorcode;

int midx, midy;

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);}

33

line(getmaxx()/2,0,getmaxx()/2,getmaxy());

line(0,getmaxy()/2,getmaxx(),getmaxy()/2);

midx = getmaxx() / 2;

setfillstyle(EMPTY\_FILL, getmaxcolor());

pieslice(midx,2, 225, 315, rad);

getch();

closegraph();

}

Arc:-

void arch::arech(){

int gdriver = DETECT, gmode, errorcode;

int midx, midy;

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

midx = getmaxx() / 2;

midy = getmaxy() / 2;

setfillstyle(EMPTY\_FILL, getmaxcolor());

arc(midx, midy, radx, rady, rad);

getch();

closegraph();

}

Rectangle:-

void rect::rec(void)

{

34

int midx,midy;

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

midx=getmaxx()/2;

midy=getmaxx()/2;

rectangle(2,4,2+radx,4+rady);

getch();

closegraph();

}

35

**APPENDIX C:**

**SOFTWARE AND HARDWARE REQUIREMENT**

**Hard ware requirements:**

* RAM : 2GB .
* Graphics card : Basic GPU.
* Hard disk : 1GB free space (or above).
* Display resolution : 1024\*768 or higher.

**Software requirements:**

* Operating system : windows 7 (or above).
* Processor : Intel Pentium
* Compiler : GCC.
* IDE : Turbo C++.
* Language :C/C++.

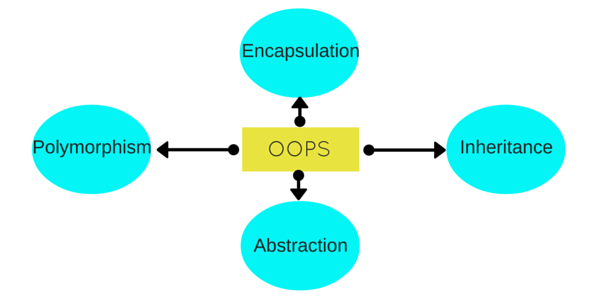
36

**APPENDIX D:**

**TECHNOLOGY USED**

C++ is a general purpose programming language that was developed by Bjarne Stroustrup at Bell labs in early 1980s as an extension to C language.

It is an object oriented programming language



37