

Assignment-05

Koch curve 1 (1st approximation)

```
#include<iostream>
#include<math.h>
#include<graphics.h>
using namespace std;
class kochCurve
{
public:
void koch(int it,int x1,int y1,int x5,int
y5)
{
int x2,y2,x3,y3,x4,y4;
int dx,dy;
if(it==0)
{
line(x1,y1,x5,y5);
}
else
{

```

```

delay(10);
dx=(x5-x1)/3;
dy=(y5-y1)/3;
x2=x1+dx;
y2=y1+dy;
x3=(int)(0.5*(x1+x5)+sqrt(3)*(y1-y5)/6);
y3=(int)(0.5*(y1+y5)+sqrt(3)*(x5-x1)/6);
x4=2*dx+x1;
y4=2*dy+y1;
koch(it-1,x1,y1,x2,y2);
koch(it-1,x2,y2,x3,y3);
koch(it-1,x3,y3,x4,y4);
koch(it-1,x4,y4,x5,y5);
}
}
};

int main()
{
    kochCurve k;
    int it;
    cout<<"enter number of iterations"<<endl;

```

```
cin>>it;
int gd=DETECT, gm;
initgraph(&gd, &gm, NULL);
k.koch(it, 150, 20, 20, 280);
getch();
closegraph();
return 0;
}
```

Explanation of the Koch Curve Program

The Koch curve is a fractal curve and one of the earliest fractals to be described. This C++ program uses recursion to draw the Koch curve using the graphics library. The program allows the user to specify the number of iterations, which determines the level of detail of the curve.

Key Components of the Code

1. Class Definition:

- The `kochCurve` class contains a method `koch()` that implements the Koch curve drawing algorithm.

2.Koch Function:

- Parameters:
- `it`: The current iteration level.
- `(x1, y1)`: Starting point coordinates.
- `(x5, y5)`: Ending point coordinates.
- The function works recursively:
- If `it` equals 0, it draws a straight line between the starting and ending points.
- Otherwise, it divides the line segment into three parts and calculates the coordinates of the new points that create the "peak" of the Koch curve.

3.Main Function:

- The program prompts the user for the number of iterations.
- It initializes the graphics system and calls the `koch()` function with the initial

points and the specified number of iterations.

- Finally, it waits for a key press before closing the graphics window.

Koch Curve 2 (2nd approximation)

```
#include<iostream>
#include<math.h>
#include<graphics.h>
using namespace std;
class kochCurve
{
public:
void koch(int it,int x1,int y1,int x5,int
y5)
{
int x2,y2,x3,y3,x4,y4;
int dx,dy;
if(it==0)
{
```

```
line(x1,y1,x5,y5);
}
else
{
delay(10);
dx=(x5-x1)/3;
dy=(y5-y1)/3;
x2=x1+dx;
y2=y1+dy;
x3=(int)(0.5*(x1+x5)+sqrt(3)*(y1-y5)/6);
y3=(int)(0.5*(y1+y5)+sqrt(3)*(x5-x1)/6);
x4=2*dx+x1;
y4=2*dy+y1;
koch(it-1,x1,y1,x2,y2);
koch(it-1,x2,y2,x3,y3);
koch(it-1,x3,y3,x4,y4);
koch(it-1,x4,y4,x5,y5);
}
}
};

int main()
```

```
{  
kochCurve k;  
int it;  
cout<<"enter number of iterations"<<endl;  
cin>>it;  
int gd=DETECT, gm;  
initgraph(&gd, &gm, NULL) ;  
k.koch(it, 150, 20, 20, 280) ;  
k.koch(it, 280, 280, 150, 20) ;  
k.koch(it, 20, 280, 280, 280) ;  
getch() ;  
closegraph() ;  
return 0;  
}
```