

```
1 // COS30008, Tutorial 3, 2022
2
3 #include "Polynomial.h"
4 #include <cmath>
5 using namespace std;
6
7
8 double Polynomial::operator()(double aX) const
9 {
10     double x = 0.0;
11     for (int i = 0; i <= fDegree; i++)
12     {
13         x += fCoeffs[i] * pow(aX, i);
14     }
15     return x;
16 }
17 // derivative is broken
18 Polynomial Polynomial::getDerivative() const
19 {
20     //Polynomial Result;
21
22     // Result.fDegree = Result.fDegree - 1;
23
24     // return Result;
25     Polynomial Result;
26     Result.fDegree = fDegree - 1;
27     for (int i = 1; i <= fDegree; i++)
28     {
29         Result.fCoeffs[i - 1] = fCoeffs[i] * i;
30     }
31     return Result;
32 }
33
34 Polynomial Polynomial::getIndefiniteIntegral() const
35 {
36     Polynomial Result;
37
38     Result.fDegree = Result.fDegree + 1;
39     for (int i = 0; i <= Result.fDegree; i++)
40     {
41         Result.fCoeffs[i+1] = fCoeffs[i] / (i+1);
42     }
43
44
45     return Result;
46 }
47
48 double Polynomial::getDefiniteIntegral(double aXLow, double aXHigh) const
49 {
```

```
50     double result = 0.0;
51     Polynomial Poly = getIndefiniteIntegral();
52     for (int i= 0; i <= Poly.fDegree; i++)
53     {
54         result += Poly.fCoeffs[i] * pow(aXHigh, i);
55         result -= Poly.fCoeffs[i] * pow(aXLow, i);
56     }
57     return result;
58 }
59
60
61     // working out, can be ignored
62     //Polynomial IndefIntegral = getIndefiniteIntegral();
63     ////this->getIndefiniteIntegral() - Result(aXHigh).getIndefiniteIntegral  ↗
64     ();
65     //return 0.0f;
66     // double widthOfRectangle = (aXHigh - aXLow) / fDegree;
67     // double area = 0.0;
68     // double heightOfRectangle = 0;
69     //
70     // for (int i = 0; i < fDegree; ++i)
71     // {
72     //     heightOfRectangle = IndefIntegral(aXLow + (i + 0.5) *  ↗
73     //         widthOfRectangle) * IndefIntegral(aXLow + (i + 0.5) * widthOfRectangle);
74     //     area += heightOfRectangle * widthOfRectangle; // find the area of  ↗
75     //     the rectangle and add it to the previous area. Effectively summing up the  ↗
76     //     area under the curve.
77     // }
```