CS211

Implementing Numerical Integration Using Function Pointers

The problem is to write a function “integrate” with prototype:

```
//FUNC represents functions of one variable that take a double as input and returns a double

typedef  double (*FUNC)(double);

 double integrate(FUNC f, double a, double b);
```

so that when it is passed a function f and bounds a and b, the call:

integrate(f, a,b) will return the value of the definite integral of f evaluated between a and b.

test integrate on the following three functions:

1. double line(double x){
   ```
           return x;
   {
   ```
2. double square(double x){
   ```
           return x*x;
   {
   ```
3. double cube(double x){
   ```
           return x*x*x;
   {
   ```

And the following main function:

```
int main(){

        cout<< “The integral of f(x)=x between 1 and 5 is: “<<integrate(line, 1, 5)<<endl;

        cout<< “The integral of f(x)=x^2 between 1 and 5 is: “<<integrate(square, 1, 5)<<endl;

        cout<< “The integral of f(x)=x^3 between 1 and 5 is: “<<integrate(cube, 1, 5)<<endl;

}
```

How does integrate work?

Inside a loop we sum up the area of rectangles with a small base (say .0001) and height f(x) for each x between a and b in increments of .0001.

When the loop terminates, we return the value of the sum.

The purpose if this assignment is to see (and implement) a very simple application of function pointers.