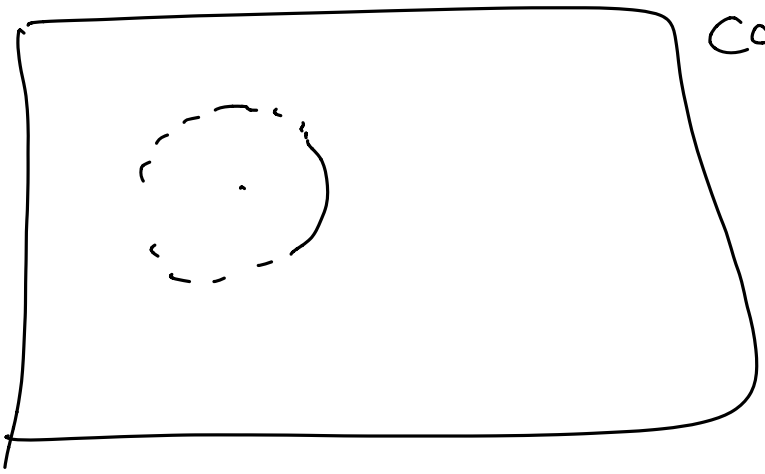
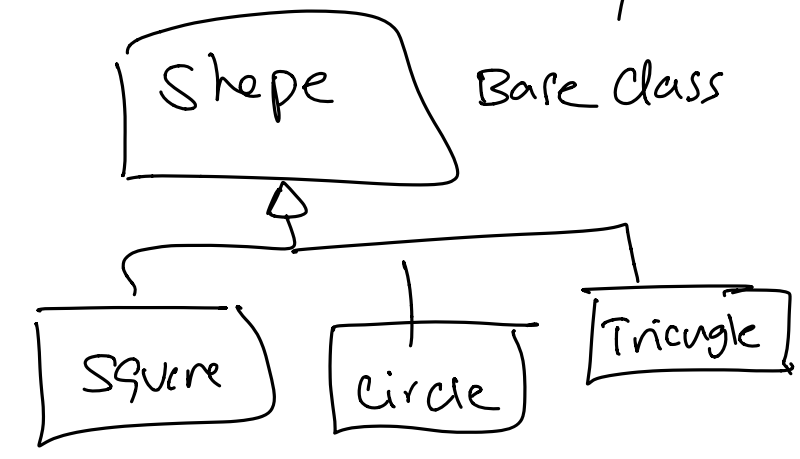
Square
Circle
Tricycle



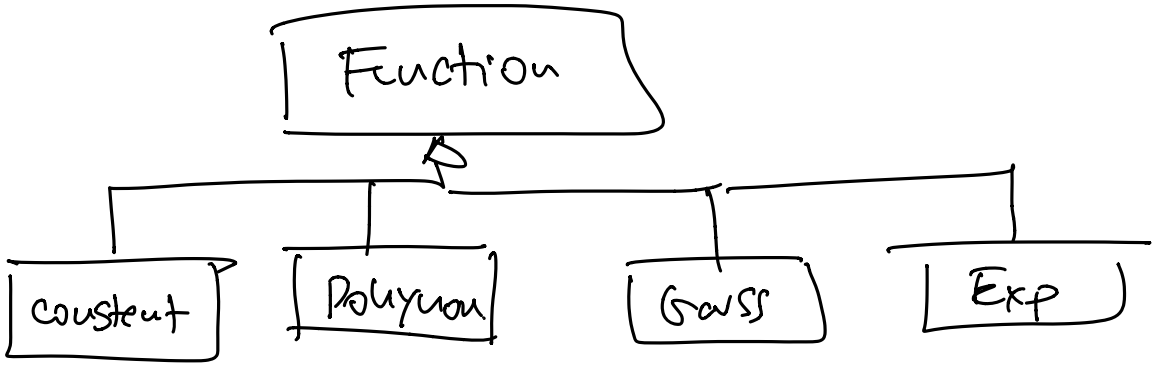



```
Class Shape {
  public:
    void draw();
  ___
  ___
};
```

```
Shape* S = { new Circle()    if i==1
           { new Square()    if i==2
           { new Triangle()  if i==3
```

S → draw()

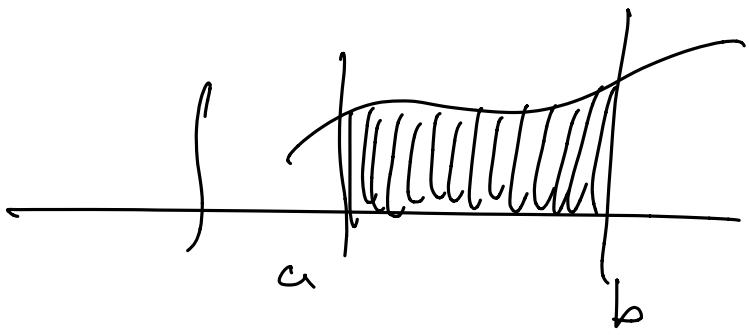
draw(): has to be different for every derived class.



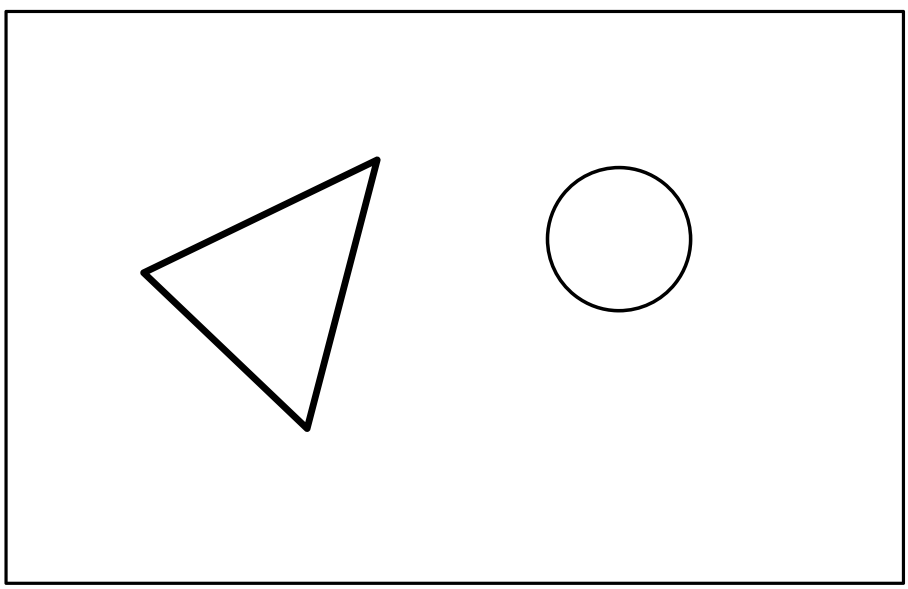
Function *f = new Gauss(0.5, 0.1)

f = new Exp(τ = -0.7)

Integration



Shape::move()



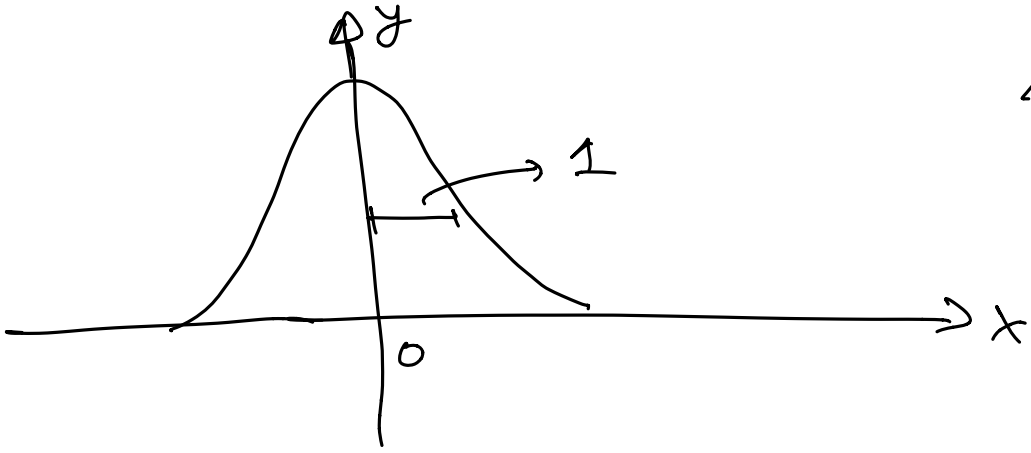
```
class Circle: public Shape {
    private:
        double radius_;
        Point2D  center_;

    void move() {
        // move center
        // draw()
    }
```

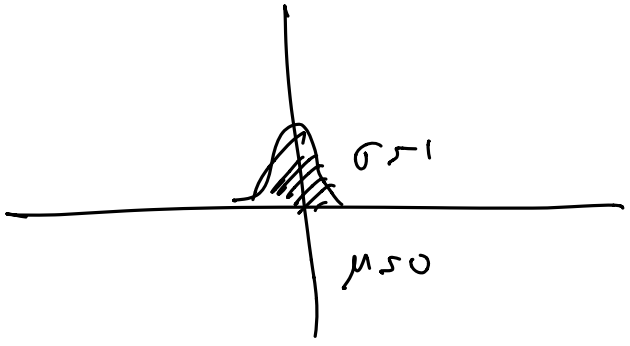
void draw() }

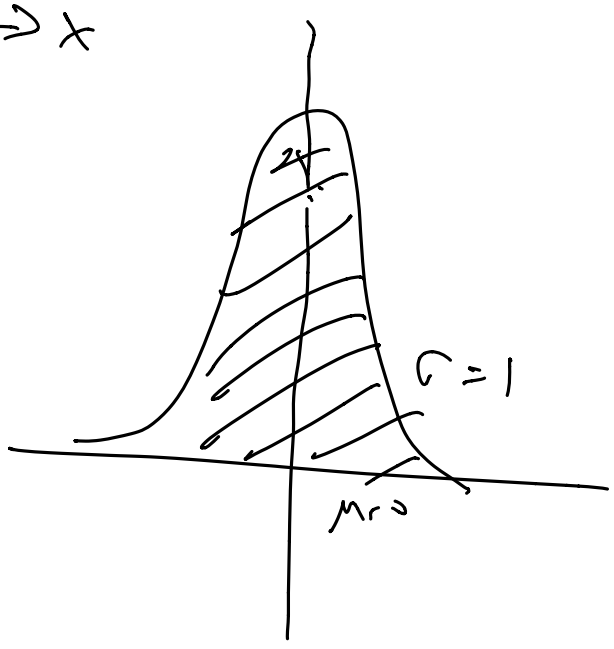
Functions

Function f = new Gauss(0,1)

$\mu = 0$ $\sigma = 1$







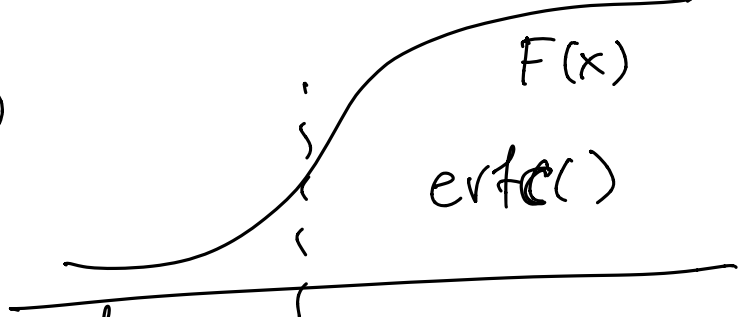
$$G(A,\mu,\sigma) = N \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} = A\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

x = 1.2;

f → value(x)

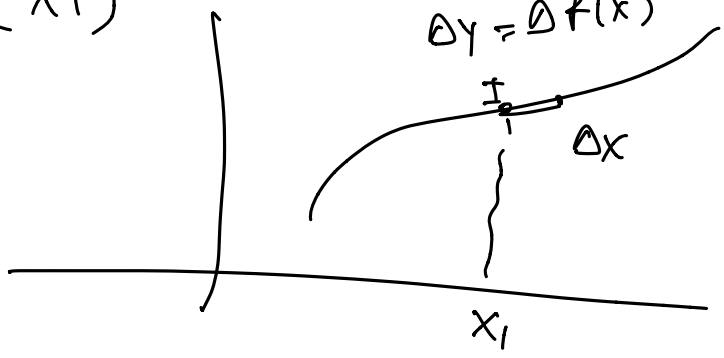
f → primitive(x)

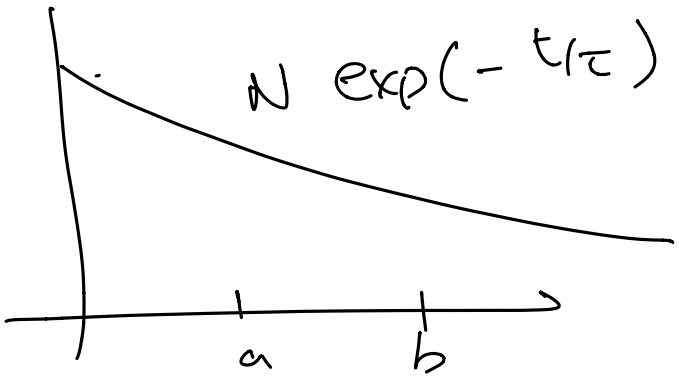
$$F(x) = \int_{-\infty}^{x} f(x')\,dx'$$



error function : primitive of Gaussian.

f → derivative (x1)



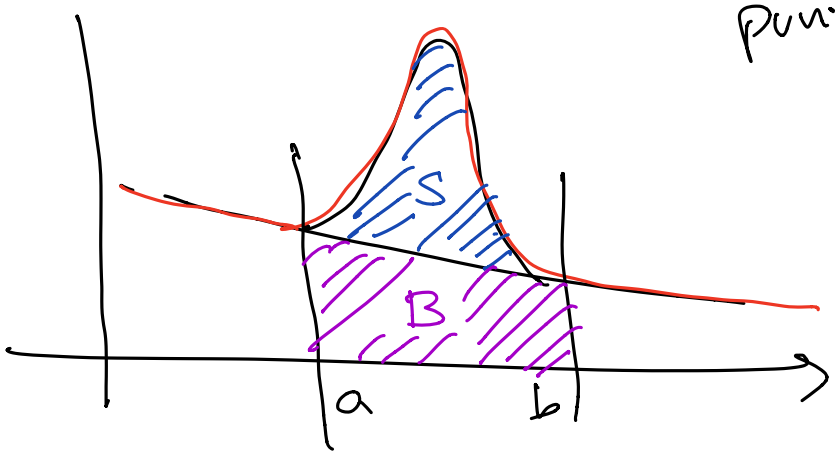


$e^{-t/\tau}$

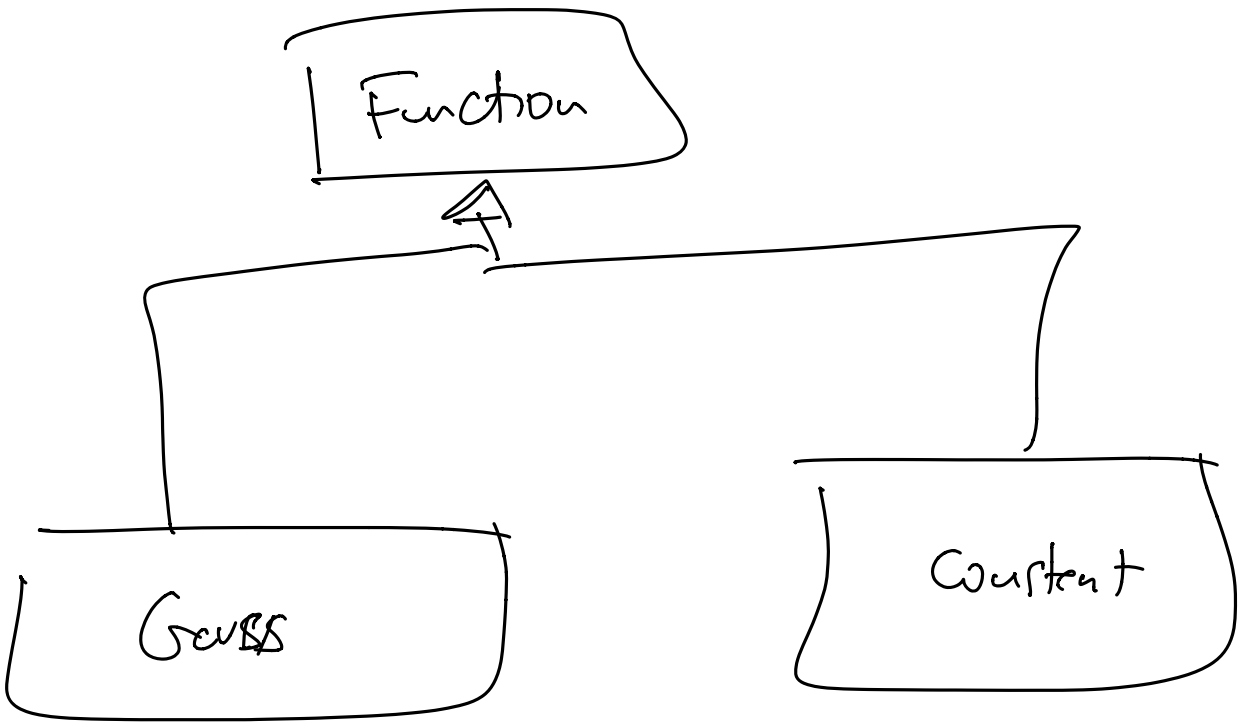
Function exp = new Exp(N,τ)

exp → integral (a, b)

$$purity = \frac{S}{B}$$



$$p = \frac{gcall \rightarrow integral(a,b)}{exp \rightarrow integral(a,b)}$$



Function.h

```
Class Function {
    publics
        double derivative(double x){
            return 0;
        }
}
```

Function.cc

```
gcc -c Function.cc
```

```
double Constant::Integral(double a, double b) const {
    return value_*(b-a);
}
```