Introduction to Modern C++

Homework 1 : Black-Scholes formula for basic option pricing

```
#include <iostream>
#include <cmath>

// Black-Scholes European Call Option Pricing
double option_price (double S, double K, double r, double sigma, double T, double t) {
    double c, d1, d2 ;

    d1 = (log(S/K)+(T-t) * (r+ sigma*sigma/2)) / (sigma*sqrt(T-t));
    d2 = d1 - sigma*sqrt(T-t);

    c = S * Phi(d1) - K * exp(-r*(T-t)) * Phi(d2);

    return c ;
}

// Vega
double Vega (double S ,double K ,double r ,double sigma ,double T ,double t) {
    double d1, vega;

    d1 = (log(S/K)+(T-t) * (r+ sigma*sigma/2)) / (sigma*sqrt(T-t));
    vega = S * sqrt(T-t) * PhiPrime(d1) ;

    return vega;
}

// Rho
double Rho (double S, double K, double r, double sigma, double T, double t ) {
    double d1, d2, rho;

    d1 = (log(S/K)+(T-t) * (r+ sigma*sigma/2)) / (sigma*sqrt(T-t));
    d2 = d1 - sigma*sqrt(T-t);

    rho = (T-t) * K * exp(-r*(T-t)) * Phi(d2);

    return rho;
}

// Delta
double Delta (double S, double K, double r, double sigma, double T, double t) {
    double d1, delta;

    d1 = (log(S/K)+(T-t) * (r+ sigma*sigma/2)) / (sigma*sqrt(T-t));
    delta = Phi(d1);

    return delta;
}

// Gamma
double Gamma (double S, double K, double r, double sigma, double T, double t) {
    double d1, gamma;

    d1 = (log(S/K)+(T-t) * (r+ sigma*sigma/2)) / (sigma*sqrt(T-t));
    gamma = PhiPrime(d1) / (S*sigma*sqrt(T-t));

    return gamma;
}
```

```
int main() {
    double S, K, r, sigma, T, t;

    std::cout << "Stock price ($) \t\t: ";
    if (!(std::cin >> S) || S <= 0) {
        std::cout << "Incorrect value of S." << std::endl;
        return -1;    // Incorrect entry, exit
    }
    std::cout << "Strike price ($) \t\t: ";
    if (!(std::cin >> K) || K <= 0) {
        std::cout << "Incorrect value of K." << std::endl;
        return -1;    // Incorrect entry, exit
    }
    std::cout << "Interest rate (-1 to +1) \t: ";
    if (!(std::cin >> r) || r < -1 || r > 1) {
        std::cout << "Incorrect value of r." << std::endl;
        return -1;    // Incorrect entry, exit
    }
    std::cout << "Volatility (0 to 1) \t\t: ";
    if (!(std::cin >> sigma) || sigma <= 0 || sigma > 1) {
        std::cout << "Incorrect value of sigma." << std::endl;
        return -1;    // Incorrect entry, exit
    }
    std::cout << "Time to maturity (in years) \t: ";
    if (!(std::cin >> T) || T <= 0) {
        std::cout << "Incorrect value of T." << std::endl;
        return -1;    // Incorrect entry, exit
    }

    t = 0;

    cout << "Option price :" << option_price(S,K,r,sigma,T,t) << std::endl;
    cout << "Delta : "       << Delta(S,K,r,sigma,T,t)        << std::endl;
    cout << "Gamma : "       << Gamma(S,K,r,sigma,T,t)        << std::endl;
    cout << "Vega  : "       << Vega(S,K,r,sigma,T,t)         << std::endl;
    cout << "Rho   : "       << Rho(S,K,r,sigma,T,t)          << std::endl;
}
```