Pricing Options by the Black-Merton-Scholes Model

Options are financial instruments that are derivatives based on the value of underlying securities such as stocks, bonds and commodity futures. An option contract offers the buyer the opportunity to buy or sell—depending on the type of contract they hold—the underlying asset.

An option is specified by a set of contractual parameters such as the following two parameters:

- Strike price (K)
- Time to maturity (T).

To price an option, one needs to introduce a model which typically involves a set of modeling parameters. The following are examples of modeling parameters.

- Risk free rate (r)
- Current price of the underlying asset (S)
- Volatility parameter of the asset price model (sigma).

You're required to design and implement an option class following the guidelines below:

1. Design a *Option* class. Its header file should include the following
   1.1. Private members:
      1.1.1. A method *init()* to initialize the option class with default parameters
      1.1.2. Parameters:
         - Strike price (K)
         - Current price of underlying (S)
         - Risk free rate (r)
         - Time to maturity (t)
         - Volatility (sigma)

   1.2. Public members:
      1.2.1. A default constructor – should call *init()* to initialize the option with default parameters
      1.2.2. A constructor with parameters as arguments to initialize the option with different parameters
      1.2.3. A destructor
      1.2.4. Get *get()* method for each of the parameters

2. Design a *pricing_method* class as an abstract class. Its header file should have the following declared as pure virtual functions with *public* access specifier
   2.1. *Black_Scholes_Option_Price()*

        2.1.1. *return* the European option price computed using black scholes analytical pricing formula

    2.2. *Binomial_Option_Price()*
        2.2.1. *return* the European option price computed using the binomial lattice method

3. Design an *Option_Price* class that derives from *Option* class and *pricing_method* class. Its header file should implement the following
    3.1. The *Option_Price* class has a public member called *flag* which is initialized using a constructor.
    3.2. *Black_Scholes_Option_Price()*
        3.2.1. Implements the black scholes analytical pricing formula for a European option
    3.3. *Binomial_Option_Price()*
        3.3.1. Computes the up and down factors and the risk neutral probabilities and implements the binomial lattice method to price a European option.
    3.4. Both the functions *get* their parameters from the *Option* class

User Input: Code should prompt the user to provide the above defined option parameters and the option flag as either “c”/”C” or “p”/”P”.
Output: Option price computed using the binomial lattice and Black Scholes model

Useful links:

https://en.wikipedia.org/wiki/Binomial_options_pricing_model
https://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model