Factorial Calculator

Create a program that keeps calculating the factorials of numbers entered by the user as long as the user types ‘y’ or ‘Y’ when being asked whether to continue.

Note: **Bold** words are output while non-bold words are input in the following console sample.

Console

```
Factorial Calculator

Enter an integer (1-10): 0

The number is not between 1 and 10!

Enter an integer (1-10): 11

The number is not between 1 and 10!

Enter an integer (1-10): 4

The factorial of 4 is 24.

Type y to continue or another letter to quit: y

Enter an integer (1-10): 5

The factorial of 5 is 120.

Type y to continue or another letter to quit: Y

Enter an integer (1-10): 6

The factorial of 6 is 720.

Type y to continue or another letter to quit: s
```

Specifications

- The exclamation point is used to identify a factorial. For example, the factorial of the number n is denoted by n!. Here’s how you calculate the factorial of the numbers 1 through 5:

```
1! = 1                  which equals 1
2! = 1 * 2              which equals 2
3! = 1 * 2 * 3          which equals 6
4! = 1 * 2 * 3 * 4      which equals 24
5! = 1 * 2 * 3 * 4 * 5  which equals 120
```

- In this program, numbers input for factorial calculation must be between 1 and 10.
- You have to define a class *Factorial* with a private data member corresponding to the number for factorial calculation.
- Declare and define a non-default constructor which requires a parameter corresponding to the private data member for the class *Factorial*.
- Declare and define a getter and a setter for the private member in the class *Factorial*.

- Declare and define a public member function *get_factorial()* in the class *Factorial* to calculate the factorial and return it as the result. You are only allowed to use loops instead of any formula.
- Declare and define a public member function *display_results()* in the class *Factorial* to display the results in the console as shown in the console sample above by calling the other member function *get_factorial()*.
- In the *main()*, after displaying the title, you must use a loop to allow the user to continue as long as the user enters 'y' or 'Y'. In each iteration of the loop, after getting the input of the number for factorial calculation from the user, you must first verify the validity of it. Only when the number passes the validity verification, a *Factorial* object will be created based on user's input and then the member function *display_results()* will be called on the object before asking the user whether to continue. Otherwise, the whole process will be started over after displaying an error message as per the console sample above.
- Your program is not supposed to ask the user whether they want to continue or not if the current input doesn't pass the validity verification.
- Assume the user will always enter integers as the input.