**Question**

A super class Number is defined to calculate the factorial of a number. Define a sub class Series to find the sum of the series S = 1! + 2! + 3 ! + 4!+ . . . . . . . . . … . ..+n!  
The details of the members of both the classes are given below:  
Class name: Number  
Data member/instance variable:  
n:to store an integer number  
Member functions/methods:  
Number(int nn):parameterized constructor to initialize the data member n=nn  
int factorial (int a):returns the factorial of a number (factorial of n = 1x 2x 3 x . . . . . . . . . x n)  
void display():displays the data members

Class name:Series  
Data member/instance variable:  
sum:to store the sum of the series  
Member functions/methods:  
Series( . . .):parameterized constructor to initialize the data members of both the classes  
void calsum( ):calculates the sum of the given series  
void display( ):displays the data members of both the classes

Using the concept of inheritance, specify the class Series giving the details of the constructor( . . .), void calsum( ) and void display( ).

**Algorithm**

Start

1. Define a class `Number` with the following instance variable:

- `n`: an integer to store an integer number.

2. Define a constructor for the class `Number`:

- Accept a parameter `nn`.

- Initialize `n` with `nn`.

3. Define a method `factorial(int a)` for the class `Number`:

- Initialize `x` to 1.

- Use a while loop to calculate the factorial of `a` by multiplying `x` by `a` and decrementing `a` until `a` is greater than 0.

- Return `x`.

4. Define a method `display()` for the class `Number`:

- Call the `factorial(int a)` method with `n` as the argument and store the result in `fac`.

- Print the factorial of `n`.

5. Define a class `Series` that extends `Number` with the following instance variable:

- `sum`: an integer to store the sum of the series.

6. Define a constructor for the class `Series`:

- Accept a parameter `nn`.

- Call the superclass constructor with `nn`.

- Initialize `sum` to 0.

7. Define a method `calsum()` for the class `Series`:

- Use a for loop to iterate from 1 to `n`.

- For each iteration, add the factorial of the current number to `sum`.

8. Define a method `display()` for the class `Series`:

- Call the superclass `display()` method.

- Print the sum of the series.

9. Define a class `Test1` with the main method:

- Create a `Scanner` object to read input from the user.

- Prompt the user to enter a number.

- Read the number entered by the user and store it in `number`.

- Create an object of the class `Series` with `number` as the argument.

- Call the `calsum()` method of the `Series` object to calculate the sum of the series.

- Call the `display()` method of the `Series` object to display the factorial and the sum of the series.

End

**Variable Description**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Purpose** |
| n | int | To store number for which factorial is to be calculated |
| sum | int | To store sum of factorials |
| number | int | Stores integer number entered by the user |
| nn | int | To initialize value of n |