Q:13:: Write a c++ program that evaluate the following function to 0.0001% accuracy :

$$sum = 1 + (1/2)^2 + (1/3)^3 + (1/4)^4 + \ldots\ldots + (1/n)^n$$

Answer:

Here, general term = $(1/n)^n$

Given accuracy = 0.0001%

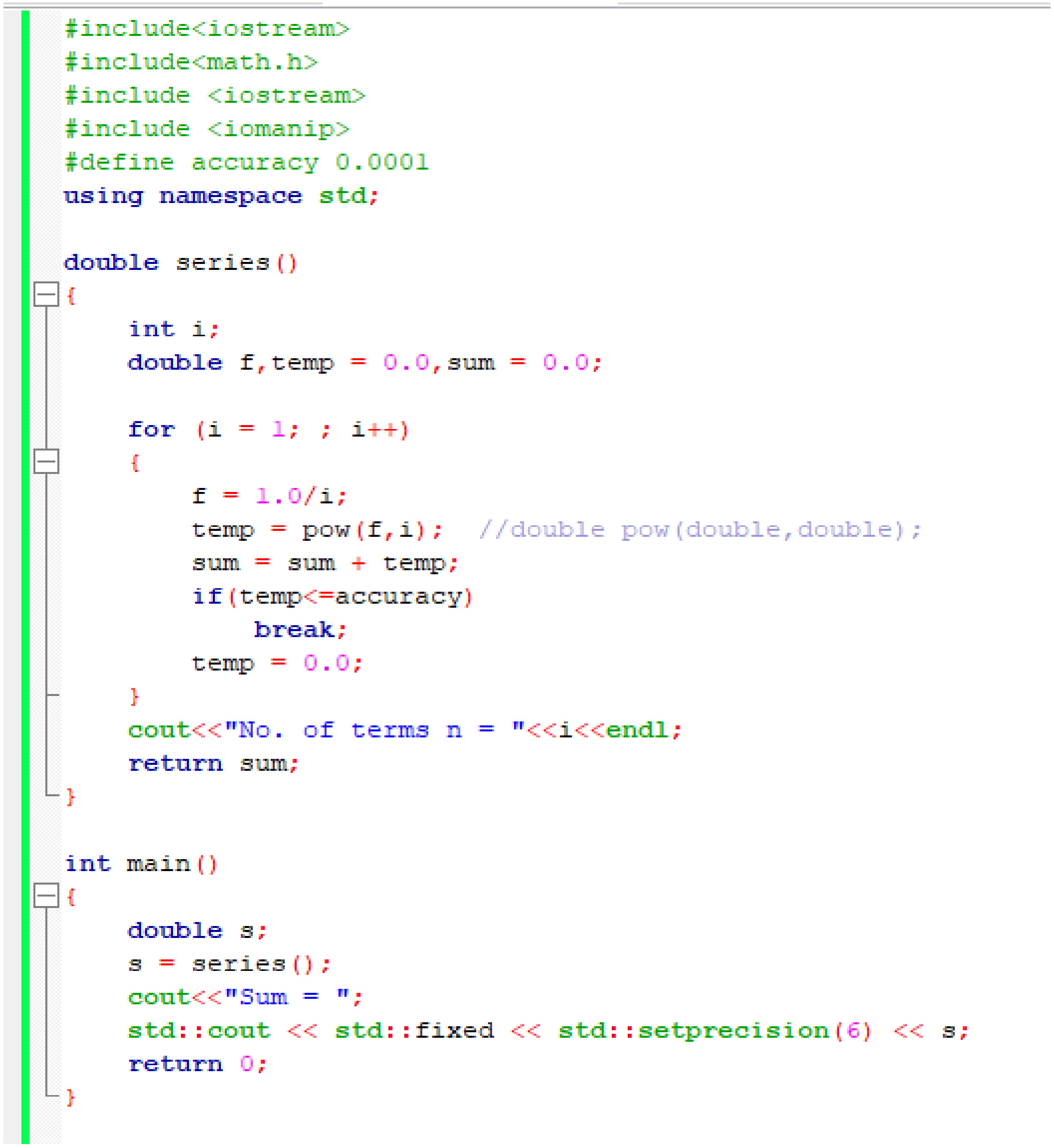
So, here n will be:

$$\begin{gathered}
(\frac{1}{n})^n = 0.0001 \\
\Rightarrow nlog(\frac{1}{n}) = log(0.0001) \\
\Rightarrow n(log1 - logn) = -4 \\
\Rightarrow n(0 - logn) = -4 \\
\Rightarrow 0 - nlogn = -4 \\
\Rightarrow -nlogn = -4 \\
\Rightarrow nlogn = 4 \\
By\ the\ method\ of\ inception \\
\Rightarrow nlogn = 6 * 10^6 \\
\Rightarrow n = 6
\end{gathered}$$

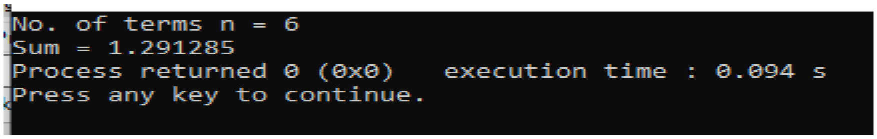
That means, we have to calculate the sum of the first 6 terms of the series.

Github code -

https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/bfa30198bb1c529ffc7fc3357cfa375df8e84b7e/chapter%204_functions%20in%20C%2B%2B/chapter%204_class/question_13_code.cpp



```cpp
#include<iostream>
#include<math.h>
#include <iostream>
#include <iomanip>
#define accuracy 0.0001
using namespace std;

double series()
{
    int i;
    double f,temp = 0.0,sum = 0.0;

    for (i = 1; ; i++)
    {
        f = 1.0/i;
        temp = pow(f,i);  //double pow(double,double);
        sum = sum + temp;
        if(temp<=accuracy)
            break;
        temp = 0.0;
    }
    cout<<"No. of terms n = "<<i<<endl;
    return sum;
}

int main()
{
    double s;
    s = series();
    cout<<"Sum = ";
    std::cout << std::fixed << std::setprecision(6) << s;
    return 0;
}
```



```
No. of terms n = 6
Sum = 1.291285
Process returned 0 (0x0)   execution time : 0.094 s
Press any key to continue.
```

Q:14:: Write a c++ program that evaluate the following function to 0.0001% accuracy :

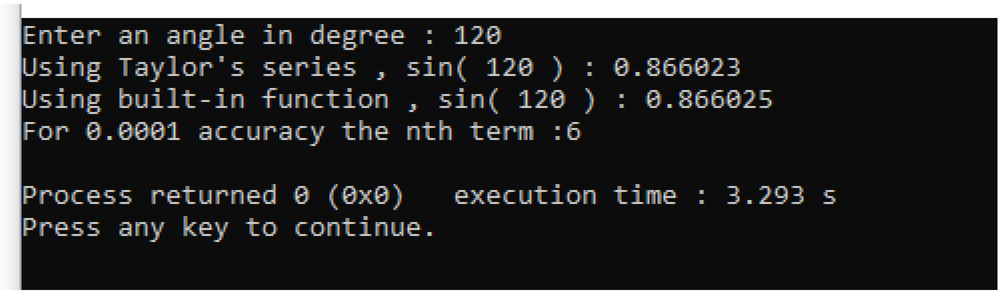
$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

or,

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$$

Answer :

```
#include<iostream>
#include<math.h>
#include<iomanip>
#define accuracy 0.0001
using namespace std;

float dg_rd(int x);
int factorial(int f);

float dg_rd(float x)
{
    float r;
    r=x*(M_PI/180);
    return r;
}
int factorial(int f)
{
    int temp=1,j=2;
    while(j<=f)
    {
        temp=temp*j;
        j++;
    }
    return temp;
}

int main()
{
    int i,sign,k=1,f;
    float x,x1,x2,sum=0.0,term;

    cout<<"Enter an angle in degree : ";
    cin>>x1;
    x2=x1;
    x=dg_rd(x2);
    for(i=1;;i=i+2)
    {
       if(k%2==0)
       {
           sign=-1;
       }
       else
       {
           sign=1;
       }
       f=factorial(i);
       term=(pow(x,i))/f;
       sum=sum+(sign*term);

       if(term<=accuracy)
        break;

       f=0;
       term=0.0;
       k++;
    }
    std::cout<<std::setprecision(6)<<"Using Taylor's series , sin( "<<x1<<" ) : "<<sum<<endl;
    cout<<"Using built-in function , sin( "<<x1<<" ) : "<<sin(x)<<endl;//radian value i.e. sin(2.0944) for sin(120*)
    cout<<"For 0.0001 accuracy the nth term :"<<k<<endl;
    return 0;
}

/*test case
sin 120 degree = sin 2.0944 radian = 0.86602540378
*/
```



```
Enter an angle in degree : 120
Using Taylor's series , sin( 120 ) : 0.866023
Using built-in function , sin( 120 ) : 0.866025
For 0.0001 accuracy the nth term :6

Process returned 0 (0x0)   execution time : 3.293 s
Press any key to continue.
```

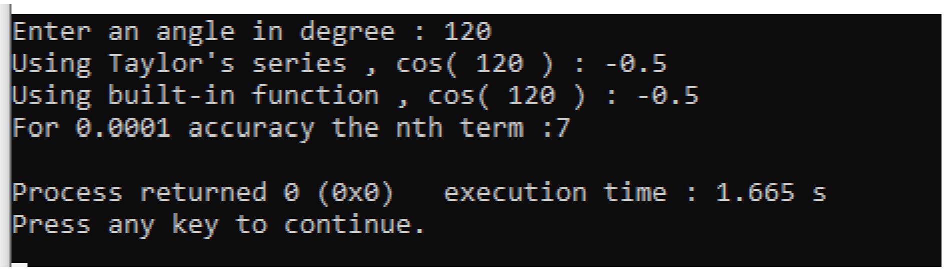
Github Code link-
https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/6064f9d84141a57d3b4ad3c12bed9f9e523829d6/chapter%204_functions%20in%20C%2B%2B/chapter%204_class/question_14_code.cpp

Q:15:: Write a c++ program that evaluate the following function to 0.0001% accuracy :

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots\ldots$$ or, $$\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

Answer :

Github Code link-
https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/6064f9d84141a57d3b4ad3c12bed9f9e523829d6/chapter%204_functions%20in%20C%2B%2B/chapter%204_class/question_15_code.cpp



```
Enter an angle in degree : 120
Using Taylor's series , cos( 120 ) : -0.5
Using built-in function , cos( 120 ) : -0.5
For 0.0001 accuracy the nth term :7

Process returned 0 (0x0)   execution time : 1.665 s
Press any key to continue.
```

```cpp
#include<iostream>
#include<math.h>
#include<iomanip>
#define accuracy 0.0001
using namespace std;

float dg_rd(int x);
int factorial(int f);

float dg_rd(float x)
{
    float r;
    r=x*(M_PI/180);
    return r;
}
int factorial(int f)
{
    int temp=1,j=2;
    while(j<=f)
    {
        temp=temp*j;
        j++;
    }
    return temp;
}

int main()
{
    int i,sign,k=1,f;
    float x,x1,x2,sum=0.0,term;

    cout<<"Enter an angle in degree : ";
    cin>>x1;
    x2=x1;
    x=dg_rd(x2);
    for(i=0;;i=i+2)
    {
       if(k%2==0)
       {
           sign=-1;
       }
       else
       {
           sign=1;
       }
       f=factorial(i);
       term=(pow(x,i))/f;
       sum=sum+(sign*term);

       if(term<=accuracy)
        break;

       f=0;
       term=0.0;
       k++;
    }
    std::cout<<std::setprecision(6)<<"Using Taylor's series , cos( "<<x1<<" ) : "<<sum<<endl;
    cout<<"Using built-in function , cos( "<<x1<<" ) : "<<cos(x)<<endl;//radian value i.e. cos(2.0944) for cos(120*)
    cout<<"For 0.0001 accuracy the nth term :"<<k<<endl;
    return 0;
}

/*test case
cos 120 degree = cos 2.0944 radian = -0.5
*/

```