

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

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

Answer:

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

Given accuracy = 0.0001%

So, here n will be:

$$\begin{aligned} \left(\frac{1}{n}\right)^n &= 0.0001 \\ \Rightarrow n \log\left(\frac{1}{n}\right) &= \log(0.0001) \\ \Rightarrow n(\log 1 - \log n) &= -4 \\ \Rightarrow n(0 - \log n) &= -4 \\ \Rightarrow 0 - n \log n &= -4 \\ \Rightarrow -n \log n &= -4 \\ \Rightarrow n \log n &= 4 \\ \text{By the method of inception} \\ \Rightarrow n \log n &= 6 * 10^6 \\ \Rightarrow n &= 6 \end{aligned}$$

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

```

#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 :

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

or,

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

Answer :

```

1  #include<iostream>
2  #include<math.h>
3  #include<iomanip>
4  #define accuracy 0.0001
5  using namespace std;
6
7  float dg_rd(float x);
8  int factorial(int f);
9
10 float dg_rd(float x)
11 {
12     float r;
13     r=x*(M_PI/180);
14     return r;
15 }
16 int factorial(int f)
17 {
18     int temp=1,j=2;
19     while(j<=f)
20     {
21         temp=temp*j;
22         j++;
23     }
24     return temp;
25 }
26
27 int main()
28 {
29     int i,sign,k=1,f;
30     float x,x1,x2,sum=0.0,term;
31
32     cout<<"Enter an angle in degree : ";
33     cin>>x1;
34     x2=x1;
35     x=dg_rd(x2);
36     for(i=1;i=i+2)
37     {
38         if(k%2==0)
39         {
40             sign=-1;
41         }
42         else
43         {
44             sign=1;
45         }
46         f=factorial(i);
47         term=(pow(x,i))/f;
48         sum=sum+(sign*term);
49
50         if(term<=accuracy)
51             break;
52
53         f=0;
54         term=0.0;
55         k++;
56     }
57     std::cout<<std::setprecision(6)<<"Using Taylor's series , sin( "<x1<<" ) :
58     "<<sum<<endl;
59     cout<<"Using built-in function , sin( "<x1<<" ) : "<<sin(x)<<endl; //radian value i.e.
60     sin(2.0944) for sin(120*)
61     cout<<"For 0.0001 accuracy the nth term : "<<k<<endl;
62     return 0;
63 }
64
65 /*test case
66 sin 120 degree = sin 2.0944 radian = 0.86602540378
67 */

```

```

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%20_functions%20in%20C%2B%2B/cha%20pter%20_class/question_14_code.cpp

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

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

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%20_functions%20in%20C%2B%2B/cha%20pter%20_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.

```

```

1  #include<iostream>
2  #include<math.h>
3  #include<iomanip>
4  #define accuracy 0.0001
5  using namespace std;
6
7  float dg_rd(float x);
8  int factorial(int f);
9
10 float dg_rd(float x)
11 {
12     float r;
13     r=x*(M_PI/180);
14     return r;
15 }
16 int factorial(int f)
17 {
18     int temp=1,j=2;
19     while(j<=f)
20     {
21         temp=temp*j;
22         j++;
23     }
24     return temp;
25 }
26
27 int main()
28 {
29     int i,sign,k=1,f;
30     float x,x1,x2,sum=0.0,term;
31
32     cout<<"Enter an angle in degree : ";
33     cin>>x1;
34     x2=x1;
35     x=dg_rd(x2);
36     for(i=0;;i=i+2)
37     {
38         if(k%2==0)
39         {
40             sign=-1;
41         }
42         else
43         {
44             sign=1;
45         }
46         f=factorial(i);
47         term=(pow(x,i))/f;
48         sum=sum+(sign*term);
49
50         if(term<=accuracy)
51             break;
52
53         f=0;
54         term=0.0;
55         k++;
56     }
57     std::cout<<std::setprecision(6)<<"Using Taylor's series , cos( "<x1<<" ) :
58     "<<sum<<endl;
59     cout<<"Using built-in function , cos( "<x1<<" ) : "<<cos(x)<<endl;//radian value i.e.
60     cos(2.0944) for cos(120*)
61     cout<<"For 0.0001 accuracy the nth term : "<<k<<endl;
62     return 0;
63 }
64
65 /*test case
66 cos 120 degree = cos 2.0944 radian = -0.5
67 */

```