namespace in c++

#include<stdio.h>

header file me sirf decleration presernt hote hai, en me function ki coding nhi hoti hai

library file me , function define kiye jate hai, yani library file me function ki coding hoti hai

toh #include me jitane me header file hoti hai , toh compiler ho us ke decleration mil jate hai . toh yeh ek traditional concept tha

c++ me 32 bit , ya 64 bit compiler me , ya en architectures me ek major change yeh kiya hai ki , jo declaration karani ki style badal diya hai , yaha pr bhi header files hoti hai , leking un header files ke andar seedhe seedhe decleration likhane ki bajaye un deceleration ko ek name diya gaya hai

yani hum kuch decleration karate hai variable ka , function ka , class ka i.e jo jo hame decleration karame hai hum kar dete hai or ese ek group me rkh dete hai

or es group ko hi hum bolate hai namespace

namespace basically decleration ka group hai , namespace ek keyword bhi hai , or namespace ki help se hum koe bhi name ka hum namespace bana sakate hai

i.e agar hum likhate hai

namespace myspace

{..........

............

}

toh myspace yaha kya hai es group ka nam

pr hum yeh likh kaha rahe hai , toh yeh hum apani source file me bhi likh sakate hai or kisi header file ko bana kr usme bhi likh sakate hai

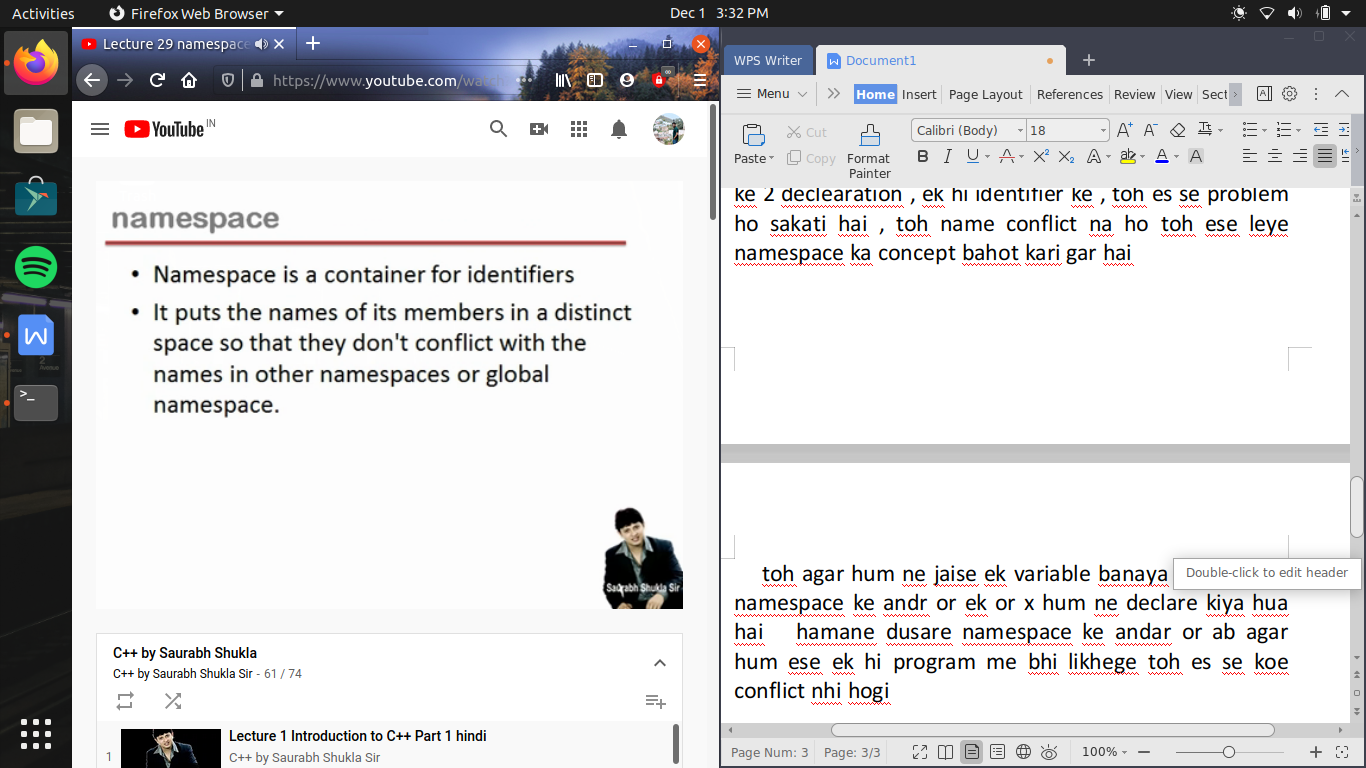
yani hum ek alag se file banaye jis me yeh namespace banayege or es ok .h extension se save kr dege toh hum ne ek header file banali jis me yeh namespace ban gya

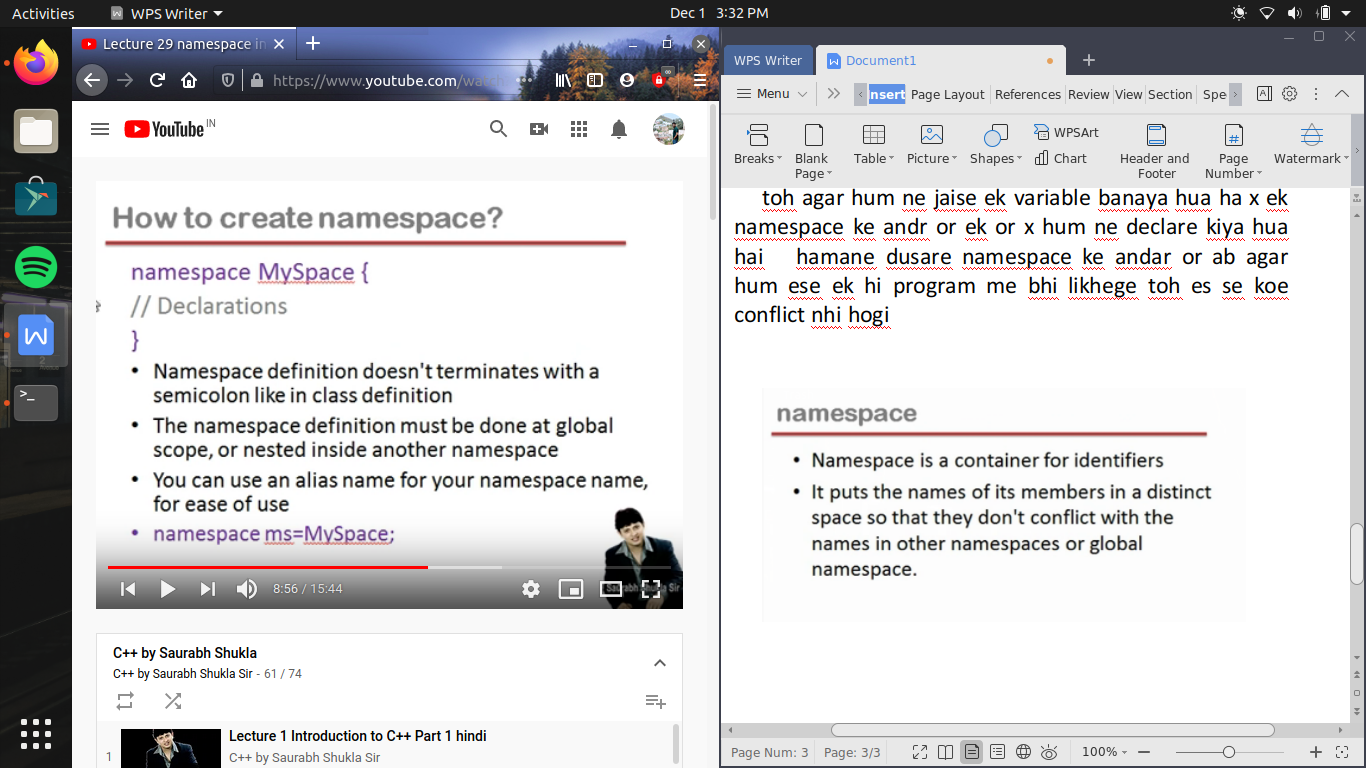
or pahale header file me directly declearation likh diya jata tha , but ab yeh namespace ka concept a gaya hai

ab namespace ke concept se fayada kya hua

toh agar ek program me 2 alag alag files jozi ja rahi hai , header files jodi ja rahi hai , or dono header file ke andar same name ke function ka ya variable ka deceleration ho chuka hai , toh fir ambiguity ho sakati hai , ya confusion ho skata hai , toh ek hi program ke 2 declearation , ek hi identifier ke , toh es se problem ho sakati hai , toh name conflict na ho toh ese leye namespace ka concept bahot kari gar hai

toh agar hum ne jaise ek variable banaya hua ha x ek namespace ke andr or ek or x hum ne declare kiya hua hai hamane dusare namespace ke andar or ab agar hum ese ek hi program me bhi likhege toh es se koe conflict nhi hogi





yeh bilkul class banane jaisa hi hai, toh jaise hum likhate hai class keyword toh yaha hum likh rahe hai namespace keyword

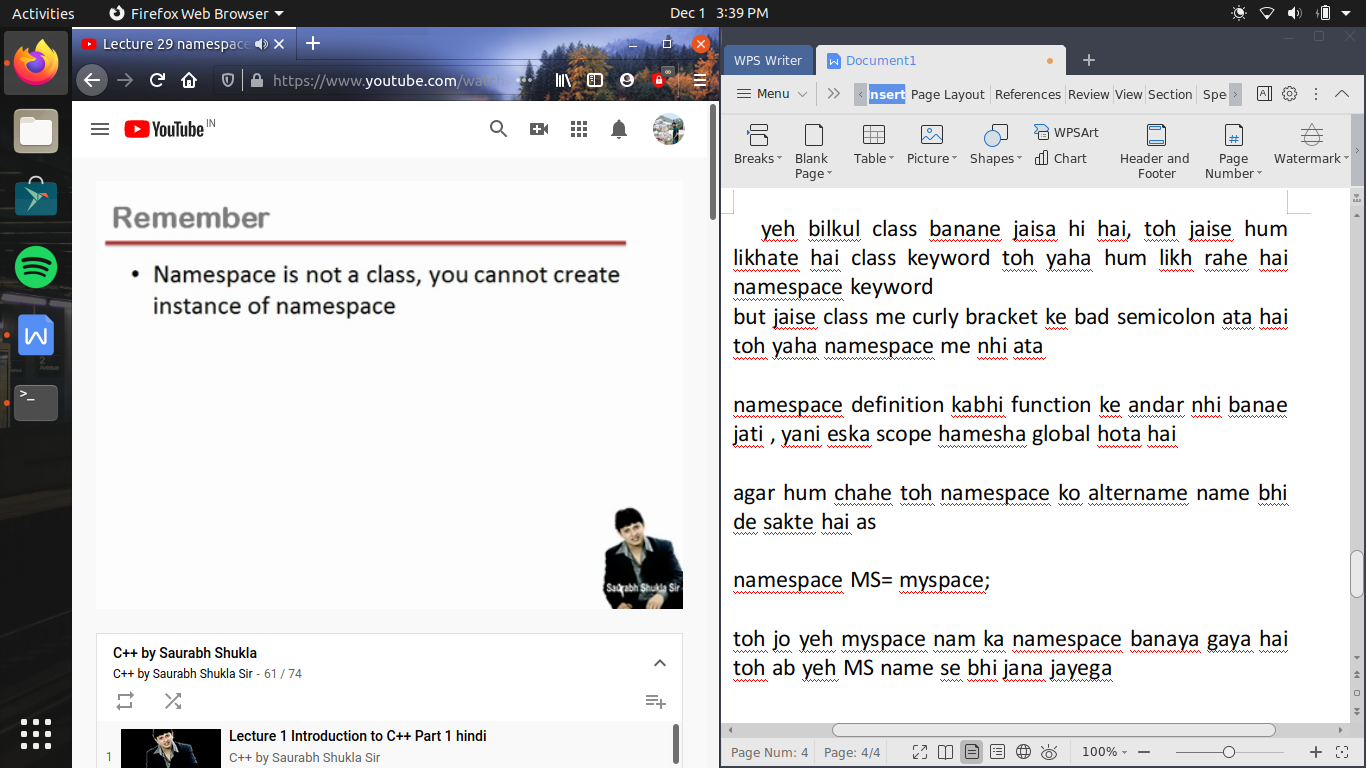
but jaise class me curly bracket ke bad semicolon ata hai toh yaha namespace me nhi ata

namespace definition kabhi function ke andar nhi banae jati , yani eska scope hamesha global hota hai

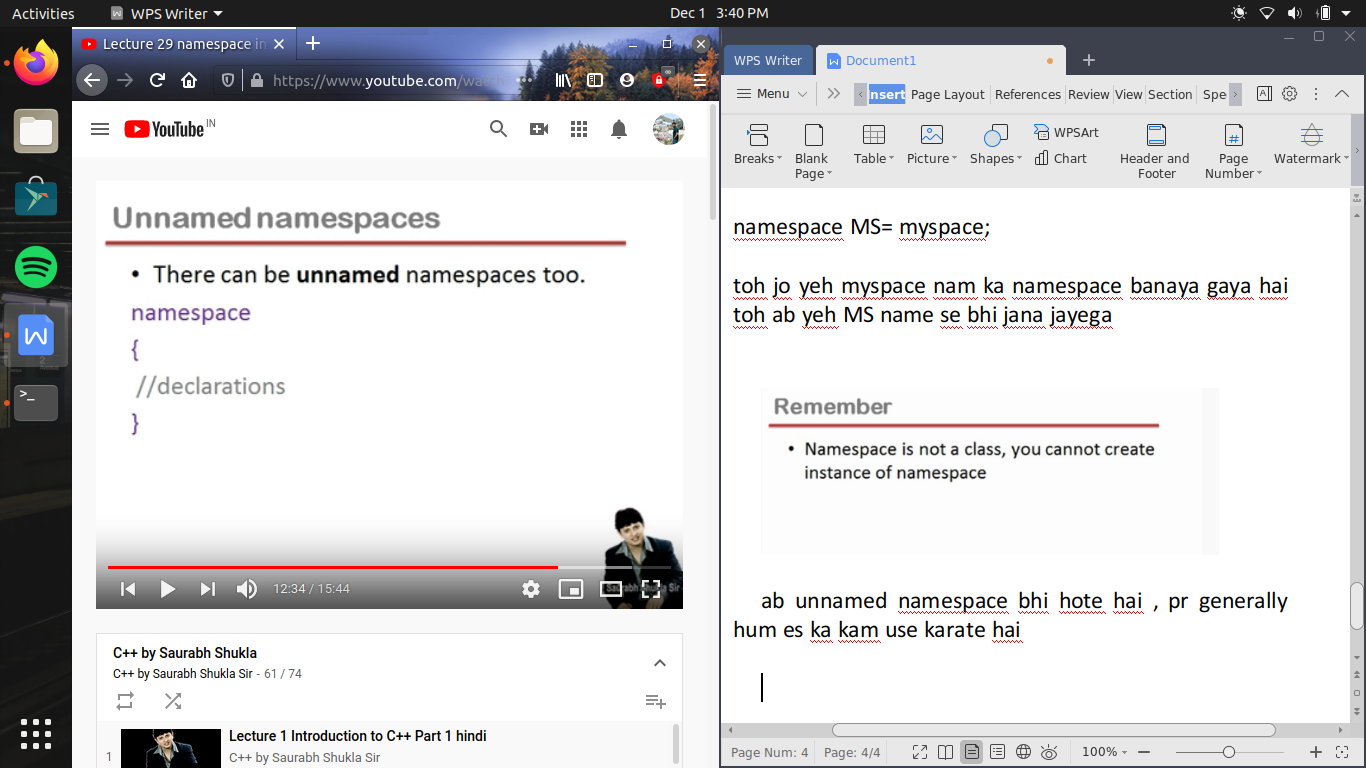
agar hum chahe toh namespace ko altername name bhi de sakte hai as

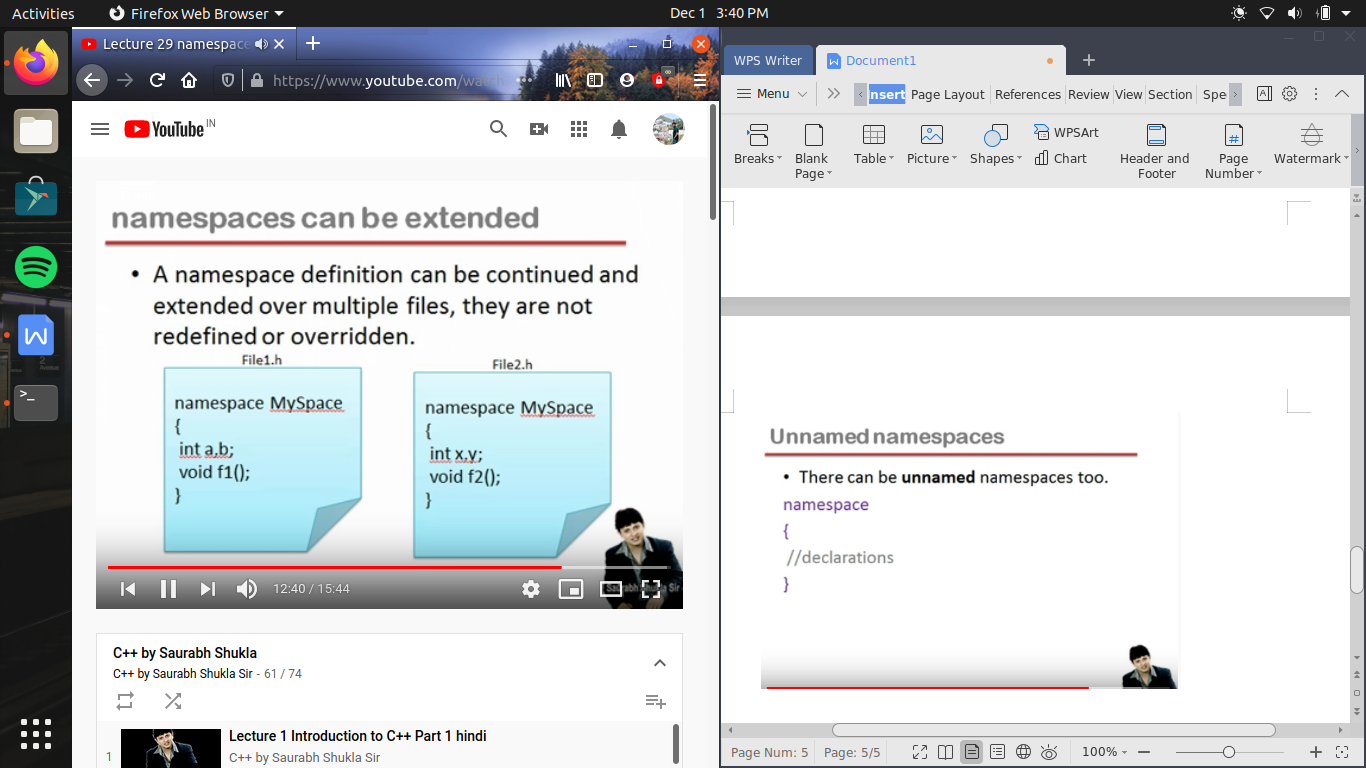
namespace MS= myspace;

toh jo yeh myspace nam ka namespace banaya gaya hai toh ab yeh MS name se bhi jana jayega



ab unnamed namespace bhi hote hai , pr generally hum es ka kam use karate hai



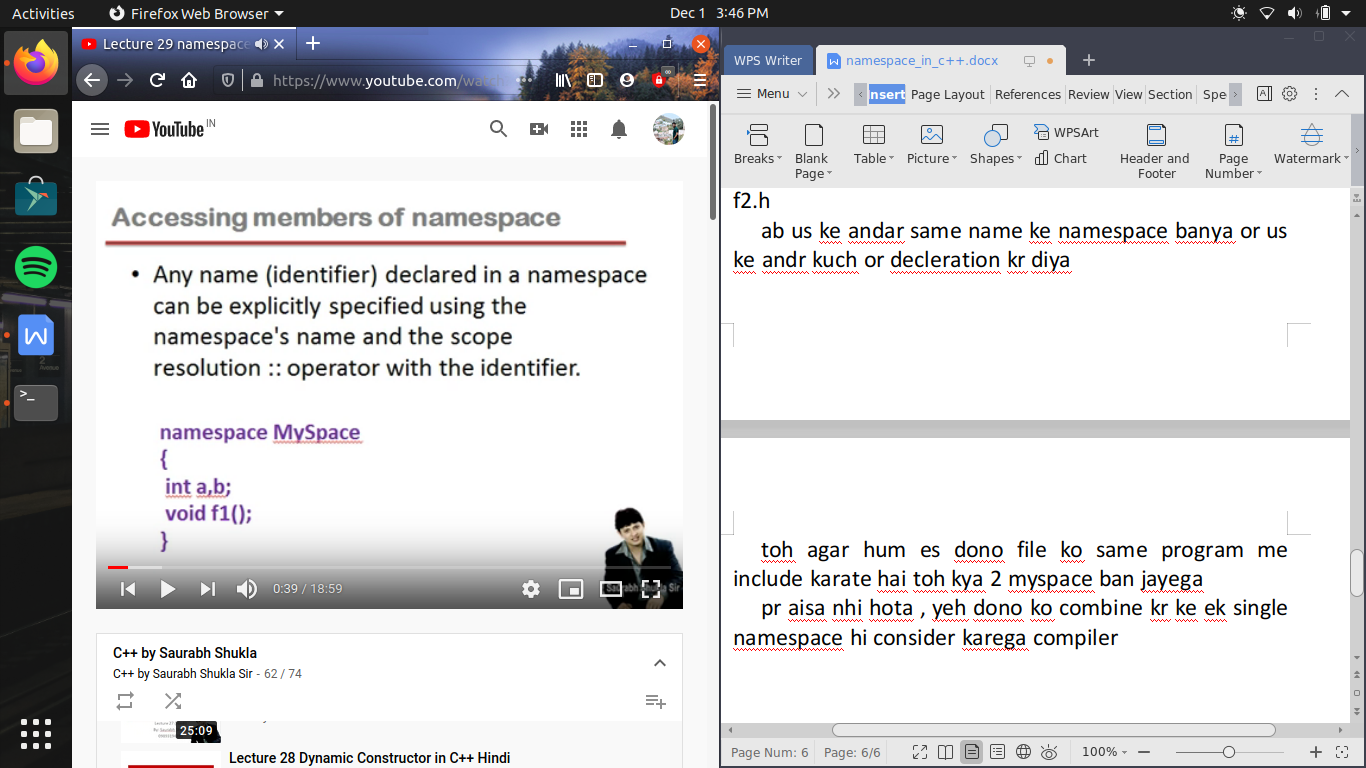


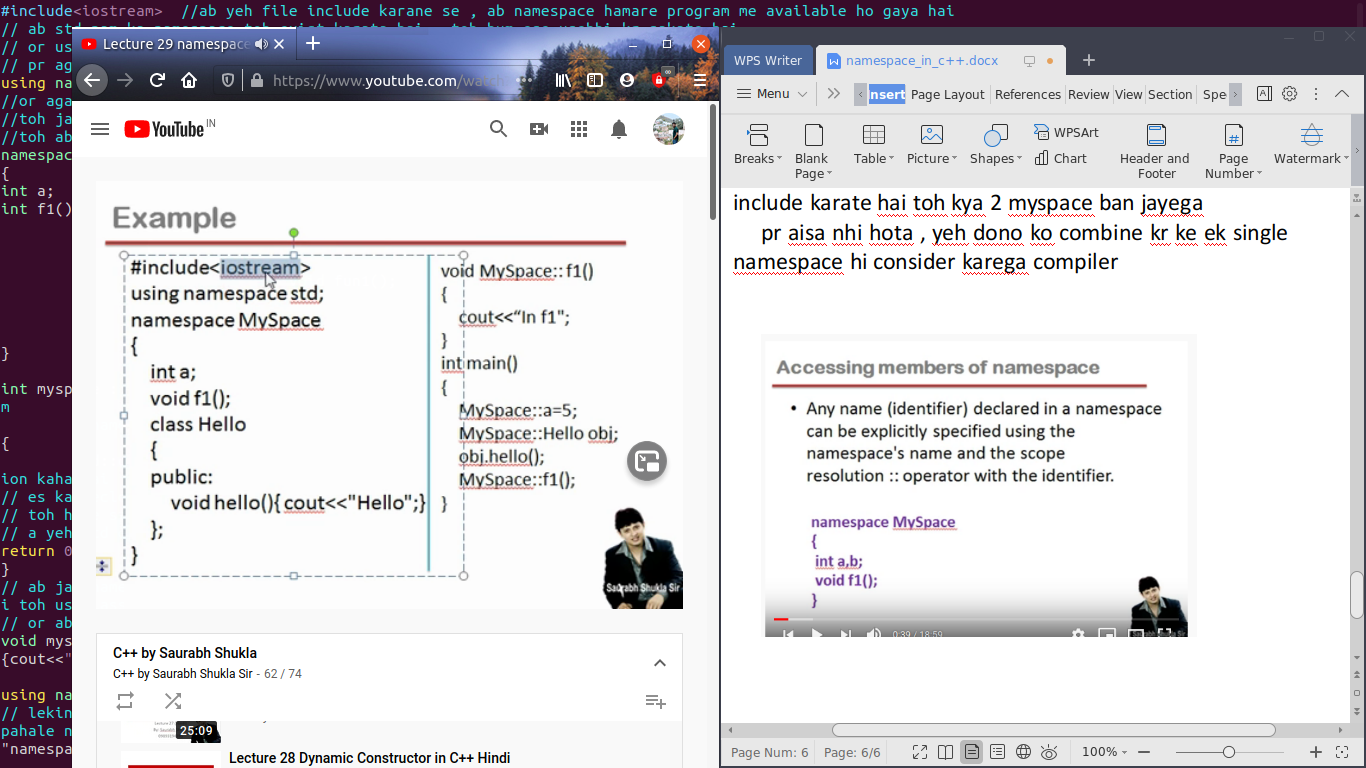
name space can be extended es ka mtlb yeh hua , ki jaise ham ne ek file banae f1.h us me hame ne ke namespace banaya myspace or us ke andar kuch decleration kr diya , ab hum ne dusari file bante hai f2.h

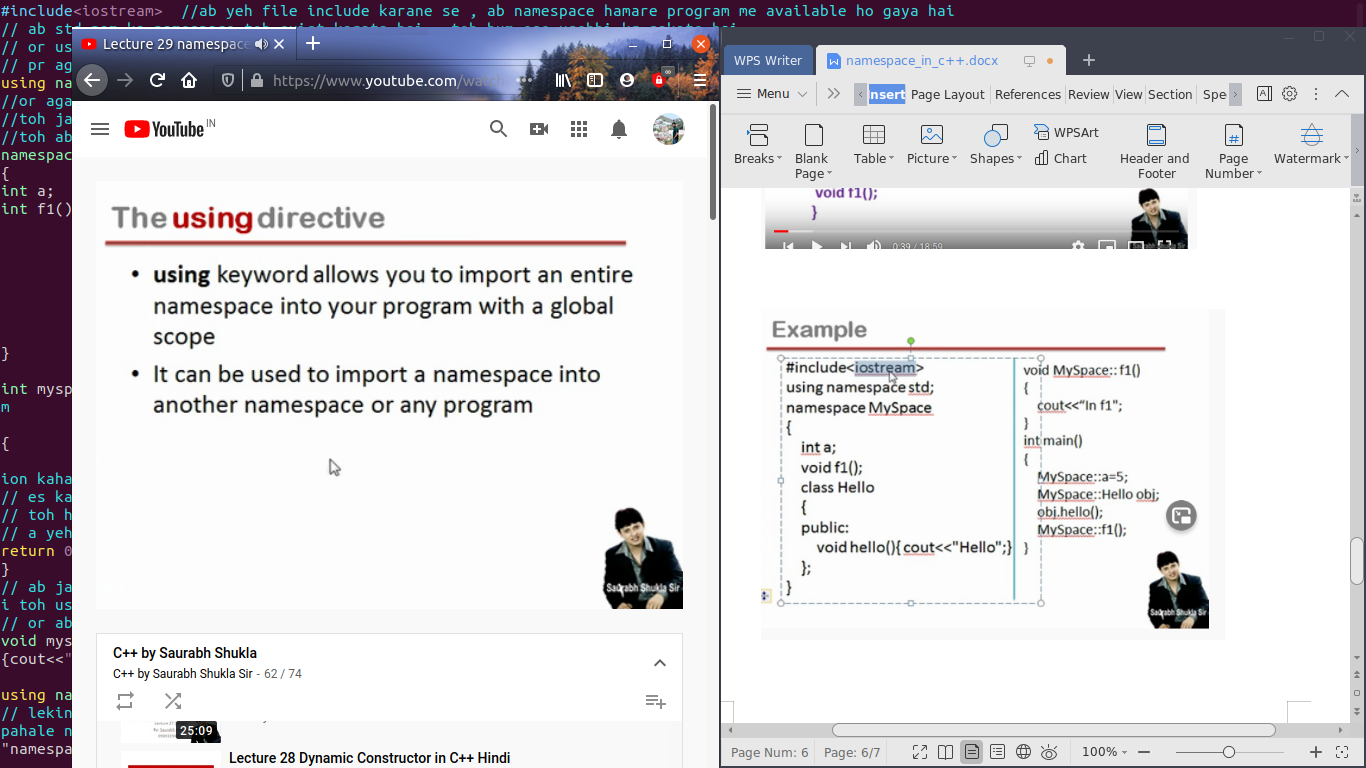
ab us ke andar same name ke namespace banya or us ke andr kuch or decleration kr diya

toh agar hum es dono file ko same program me include karate hai toh kya 2 myspace ban jayega

pr aisa nhi hota , yeh dono ko combine kr ke ek single namespace hi consider karega compiler

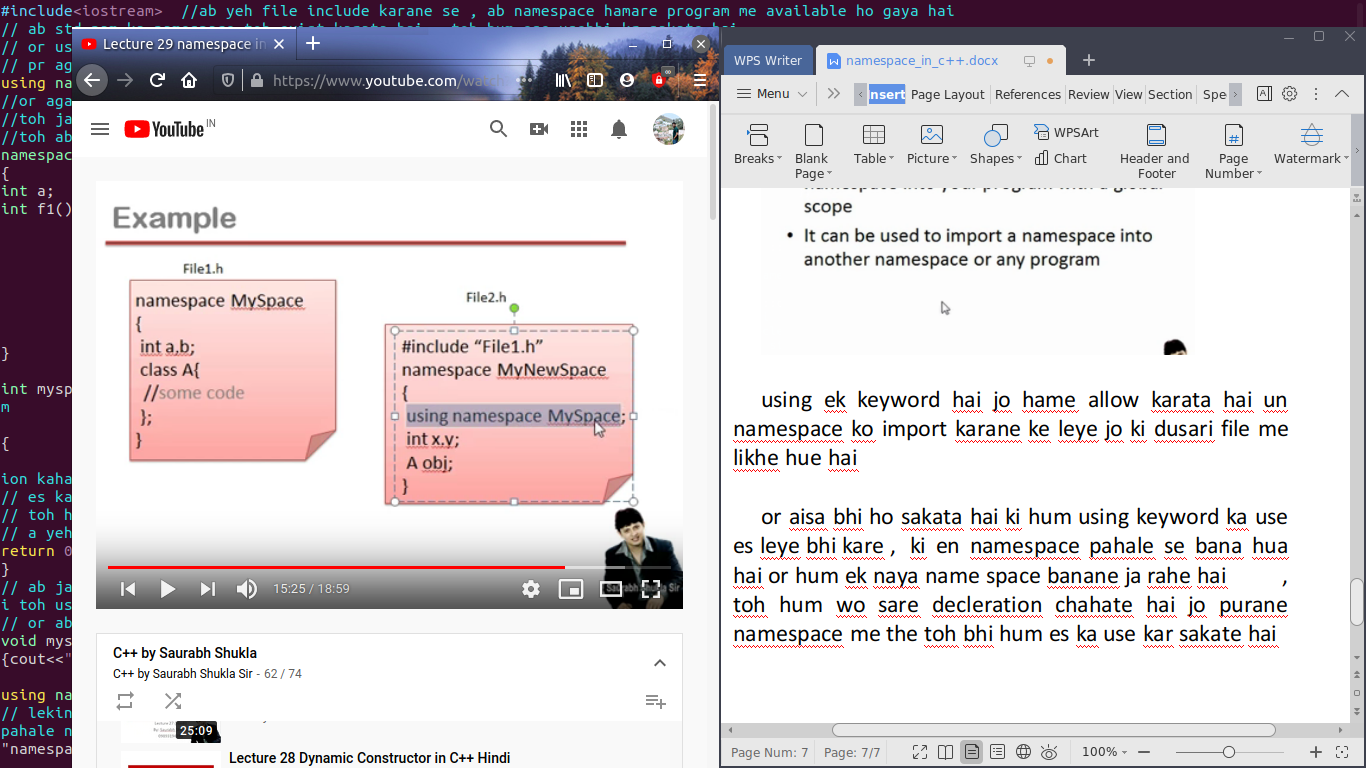






using ek keyword hai jo hame allow karata hai un namespace ko import karane ke leye jo ki dusari file me likhe hue hai

or aisa bhi ho sakata hai ki hum using keyword ka use es leye bhi kare , ki en namespace pahale se bana hua hai or hum ek naya name space banane ja rahe hai , toh hum wo sare decleration chahate hai jo purane namespace me the toh bhi hum es ka use kar sakate hai

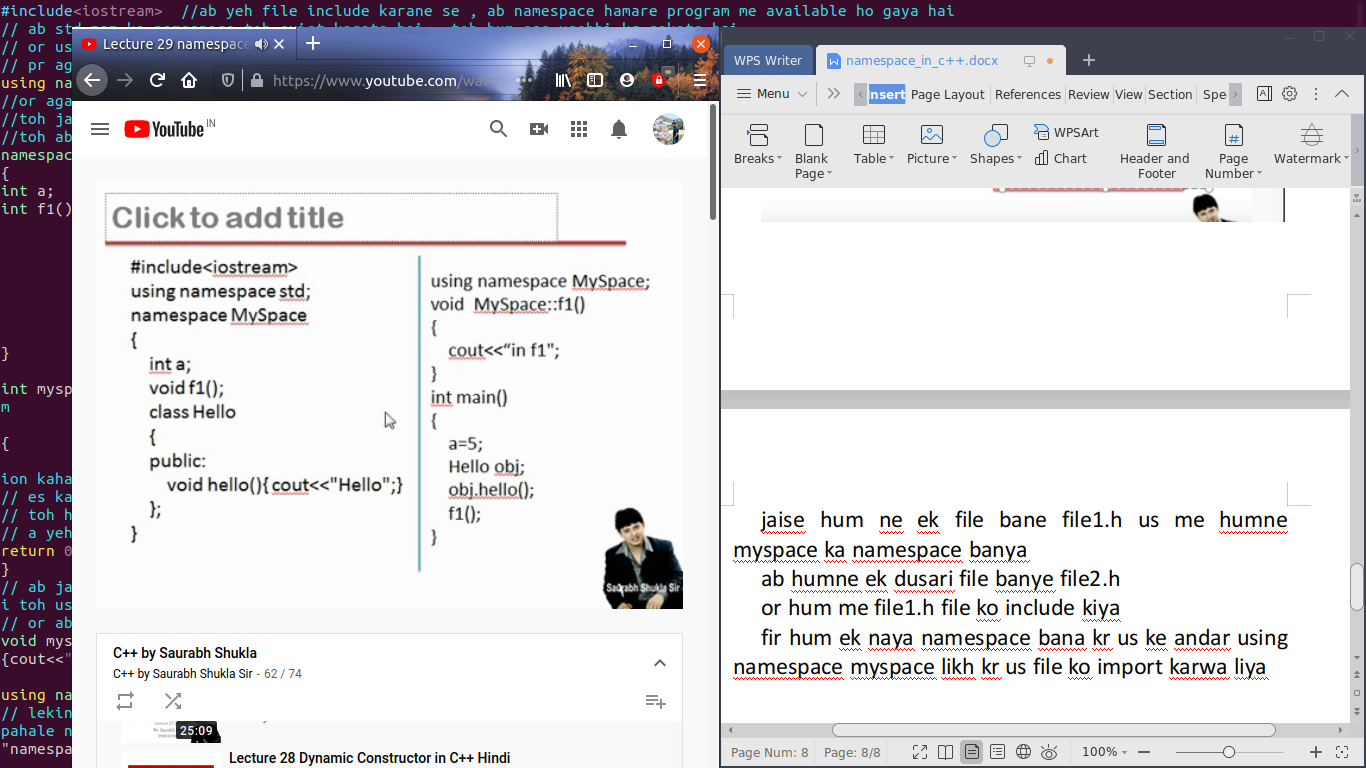


jaise hum ne ek file bane file1.h us me humne myspace ka namespace banya

ab humne ek dusari file banye file2.h

or hum me file1.h file ko include kiya

fir hum ek naya namespace bana kr us ke andar using namespace myspace likh kr us file ko import karwa liya



#include<iostream> //ab yeh file include karane se , ab namespace hamare program me available ho gaya hai

// ab std nam ka namespace toh exist karata hai , toh hum ese usebhi kr sakate hai

// or use use kaise karana hooga jaise hum myspace likh rahe hai , a use karate time

// pr agar hum asie nhi lihana chahte toh hum likate hai using namespace std;

using namespace std;

//or agar sirf hum yeh line likh de pr iostream wali file ko include na karwaye

//toh jab hum hum es file ko include nhi karege tab tak std name ka namespace ayega hi nhi program me , wo code hi nhi ayega

//toh ab es line ke bad es namespace me declare function ko use karane ke leye hame bar bar namespace ka nam likhae ki jarurat nhi padegi

namespace myspace

{

int a;

int f1();

class A

{

public:

void fun1();

};

}

int myspace ::f1() //yaha hum ne f1() define toh kiya pr yeh kaise pata chalega ki yeh wo f1() hai jo myspace ke andar declare hai toh es ke leye hum

//namespace ke name or scope resolution opeartor lagate hai ::

{

std::cout<<"HELLO F1"<<endl; // ab yaha cout bhi toh predefined object hai toh ese bhi toh declearation ki jaruarat hai , toh es ka declearation kaha hai

// es ka decleration bhi toh ek namespace me hoga , toh wo name space ka name hai std;

// toh hame std ko include karana padega

// a yeh std ek file me bana hua hai , jis ka nam hai iostream (yaha iostream.h nhi bol rahe hai )

return 0;

}

// ab jab hame fun1() define karana hai hai toh, yeh ek class ke andar hai or agar koe function class ke andr declare hota hai or bahar define hota hai toh us class ka nam or scope resolution opearaor se hum ese class ka member function bana saket hai

// or ab kyu ki hame ese myspace ka part bhi banana hai toh ek us me hum namespace ka name or scope resolution laga dete hai

void myspace::A::fun1()

{cout<<"HELLO FUN1"<<endl;}

using namespace myspace; //toh agar hum myspace ke sath bhi karate hai toh hame bar bar my space likhane ki jaruart nhi hai

// lekin agar koe ek same function ka nam dusare namespace ke andar bhi define hai toh yeh zaruri ho jata hai ki us function ko call karte time us ke pahale namespace ka name or scope resolution lagaya jaye ::

//

int main()

{

a=5;

f1();

A a1;

a1.fun1();

// myspace:: a=5;

// a=5; //ab aagar hum aisa lihate hai toh es me ayegi error yeh galat hai kyu ki a ka jo scope hai wo namespace ke nam se hi accessible hoga

// yani jo a hai wo main function ka member nhi hai ,mtlb main function ke andar nhi bana hai

// ya a koe global varibale bhi nhi hai

// yani a ka apana ek limited scope hai , or us scope ke andar hi wo access hoga

// toh yaha pr hame kya karana hota hai ki hame pahale us scope ka nam likhana hota hai myspace or us ke bad hame scope resolution operator lagana hota hai toh scope ko resolve karege

// aisa kr ke hum a ko access kr payege

//ki kahana ka mtlb yeh hai ki jo bhi upar decleration kiye gaye hai yeh sare ke sare ke group me hai or us ko hum ek scope bol rahe hai

//or us ka nam hai myspace

//or agar hame es group ke kisi bhi member ko access karana hai toh hame es group ka nam likhana padaega or fir scope resolution opearator ko use karege

//jis se ki hum scope ko resolve karege yani us limitation ke bahar hum ese use kr sakate hai

//or agar hame ese tarah function ko call karana hai

//pr function ko call kr ke koe fayada hi nhi hai jab tak wo define na ho

}

