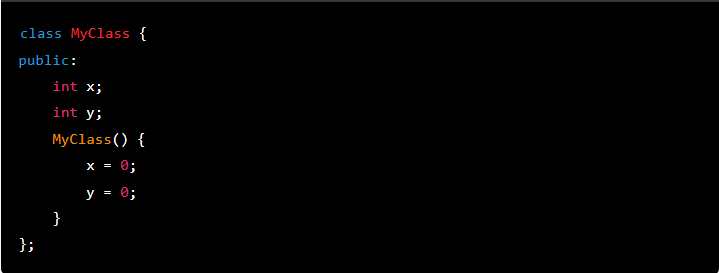
**Konstruktorlar va destruktorlar. A’zo o‘zgaruvchilarni initsializatsiyalash. Sinf a’zolarining statik o‘zgaruvchilarini aniqlash va initsializatsiyalash. Statik metodlar**

1. **Konstruktorlar va destruktorlar.**

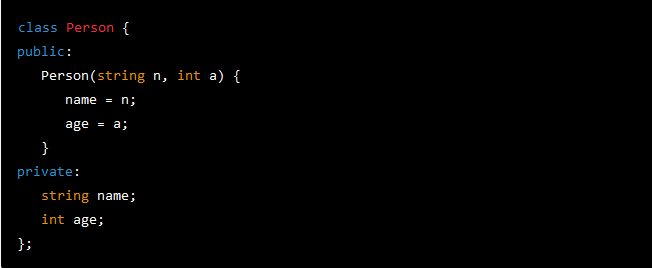
C++ tilidagi konstruktorlar sinf ob’ektlarini ishga tushirish uchun ishlatiladigan maxsus a’zo funksiyalardir. Ob’ekt yaratilganda ular avtomatik ravishda chaqiriladi va ularning nomi har doim sinf nomi bilan bir xil bo‘ladi. Quyida konstruktorlar va ulardan C++ da foydalanishga misollar keltirilgan:

**Misol uchun quyidagi kodlarni ko‘rib chiqamiz:**

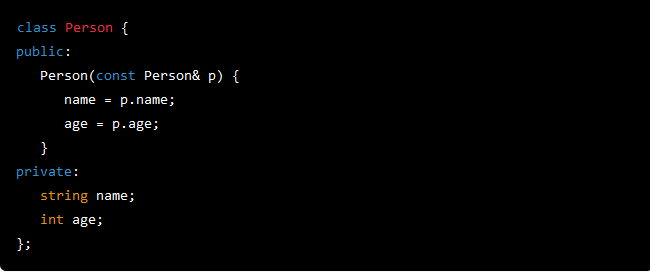
1. Standart konstruktor: Bu konstruktor argumentlarga ega emas va ob'ekt hech qanday argumentlarsiz yaratilganda avtomatik ravishda chaqiriladi. Ob'ektni standart qiymatlar bilan ishga tushiradi.



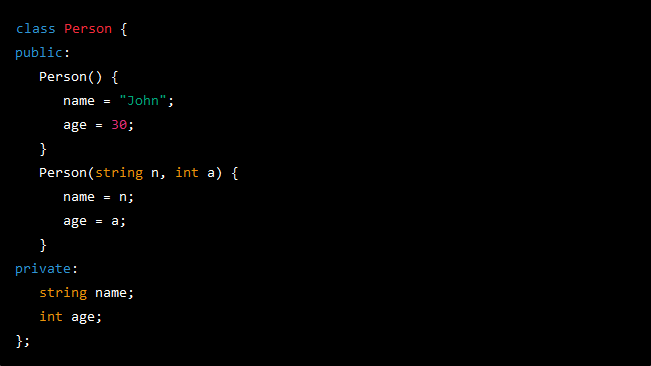
1. Parametrlangan konstruktor: Bu konstruktor bir yoki bir nechta argumentlarni oladi va ob’ektni muayyan qiymatlar bilan ishga tushirish uchun ishlatiladi.



1. Nusxalash konstruktori: Ushbu konstruktor mavjud ob’ektning nusxasi bo‘lgan yangi ob’ektni yaratadi.



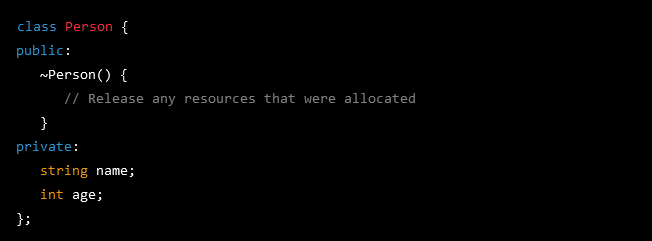
1. Konstruktorni haddan tashqari yuklash: Bu bir xil nomli, ammo turli argumentlarga ega bo‘lgan bir nechta konstruktorlarni yaratish jarayonidir. Bu foydalanuvchiga turli argumentlar to‘plamidan foydalangan holda ob’ektlar yaratish imkonini beradi.



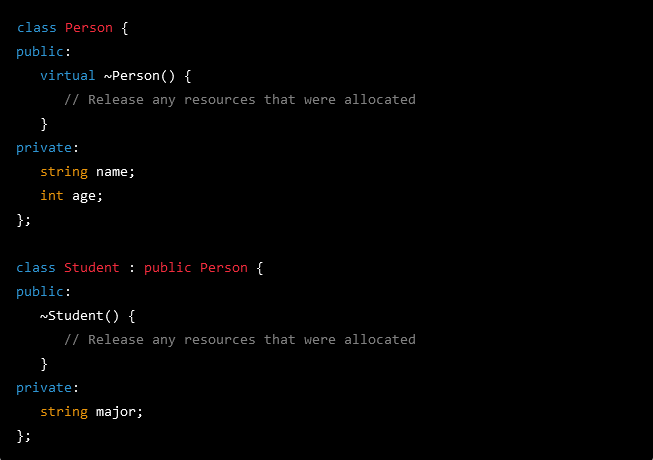
C++ tilidagi destruktorlar maxsus aʼzo funksiyalar boʻlib, obʼyekt qoʻllanish doirasidan chiqib ketganda yoki aniq oʻchirilganda avtomatik ravishda chaqiriladi. Ularning nomi sinf nomi bilan bir xil, lekin uning oldida tilda (~) belgisi mavjud.

C++ da destruktorlar va ulardan foydalanishga misollar:

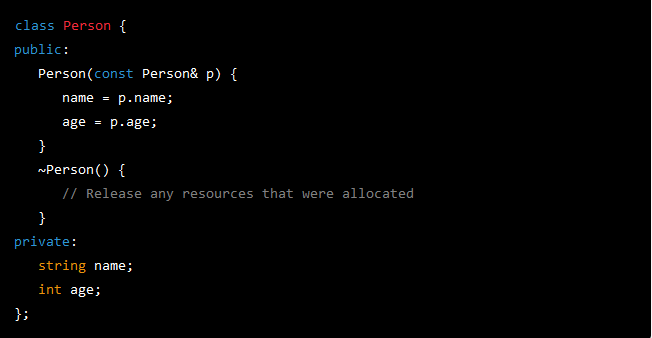
1. Standart destruktor: Bu destruktorda argumentlar yoʻq va obʼyekt qoʻllanish doirasidan chiqib ketganda yoki aniq oʻchirilganda avtomatik ravishda chaqiriladi. U ob'ekt tomonidan ajratilgan har qanday resurslarni chiqaradi.



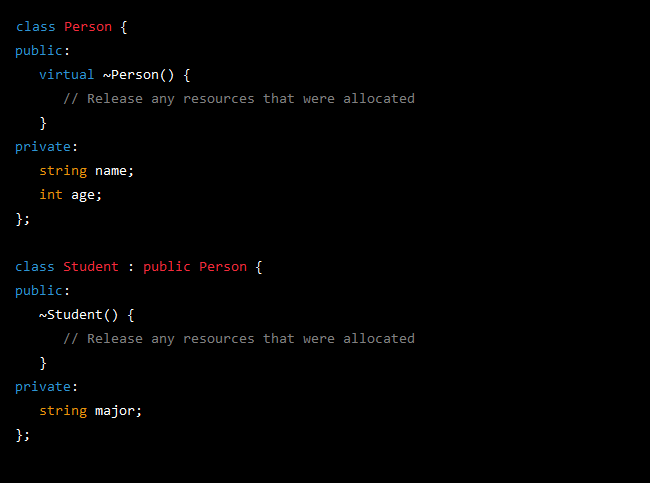
1. Virtual destruktor: Bu destruktor sinf virtual funktsiyalarga ega bo‘lganda ishlatiladi va barcha resurslar to‘g‘ri chiqarilishini ta’minlash uchun qurilishning teskari tartibida chaqiriladi. U asosiy sinfda virtual deb e’lon qilingan.



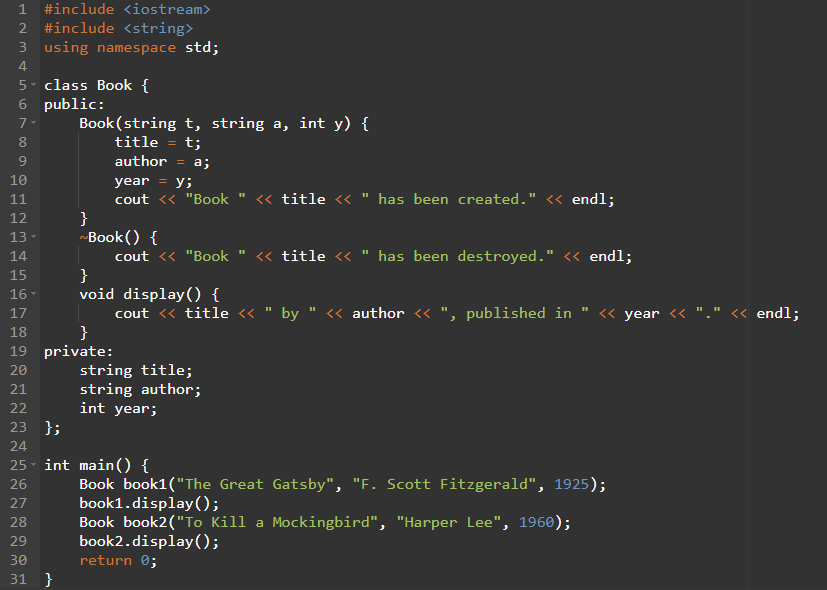
1. Nusxalash konstruktori va destruktori: Agar sinfda nusxa ko‘chirish konstruktori mavjud bo‘lsa, ko‘pincha ajratilgan har qanday resurslarni to‘g‘ri chiqaradigan destruktorni aniqlash kerak bo‘ladi. Buning sababi shundaki, nusxa ko‘chirish konstruktori ob’ekt yo‘q qilinganda o‘chirilishi kerak bo‘lgan yangi resurslarni yaratishi mumkin.

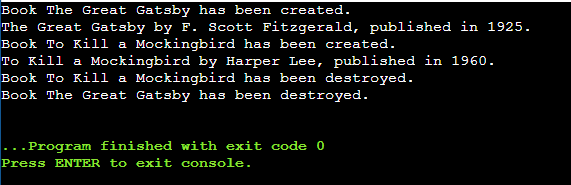


1. Destruktorni bekor qilish: Bu asosiy sinf destruktori bilan bir xil nomga ega, lekin har xil xatti-harakatlarga ega yangi destruktor yaratish jarayoni. Bu olingan sinf asosiy sinfda ajratilmagan qo‘shimcha resurslarni chiqarishi kerak bo‘lganda foydalidir.



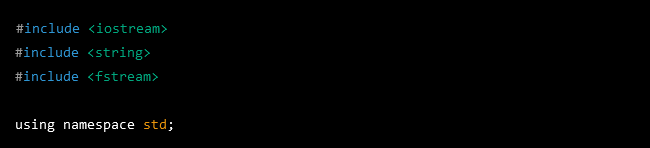
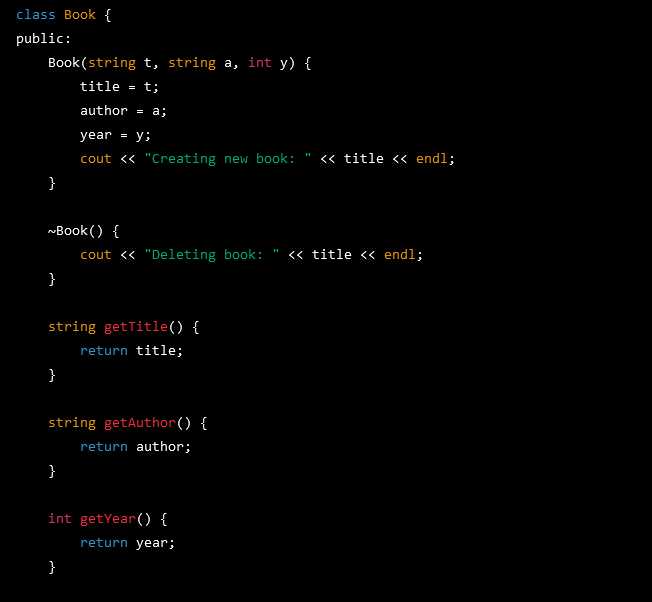
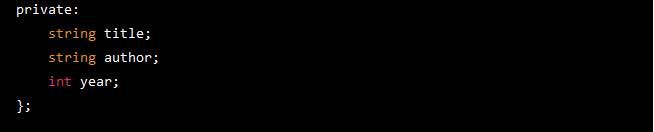
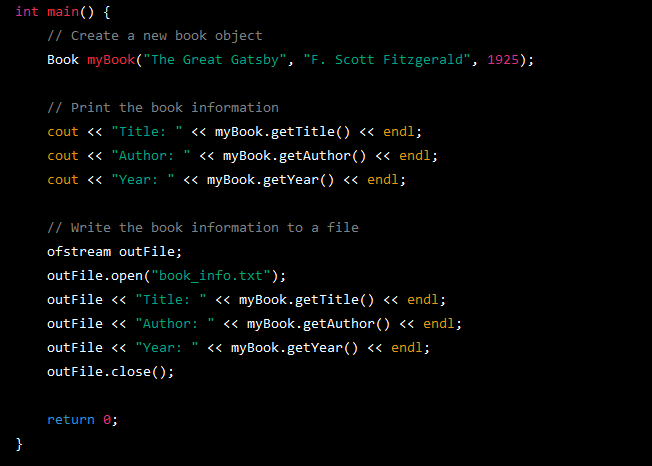
Endi konstruktor va destruktordan foydalanib kutubxona uchun dastur yaratamiz.





Ushbu dasturda Book sinfida uchta argument (nom, muallif, yil) qabul qiluvchi konstruktor va ob’ekt yo‘q qilinganda xabarni aks ettiruvchi destruktor mavjud. Displey funksiyasi kitob tafsilotlarini chop etish uchun ishlatiladi. Asosiy funktsiyada konstruktor yordamida ikkita Book ob’ekti yaratiladi va ularning tafsilotlari displey funktsiyasi yordamida ko‘rsatiladi. Dastur tugagach, har bir ob’ekt uchun avtomatik ravishda destruktor chaqiriladi, bu ob’ekt yo‘q qilinganligini bildiruvchi xabarni ko‘rsatadi.

Endi C++ dasturlash tilida konstruktor va destruktor yordamida kutubxona dasturini yarating va uni faylga yozuvchi dasturni yaratamiz.

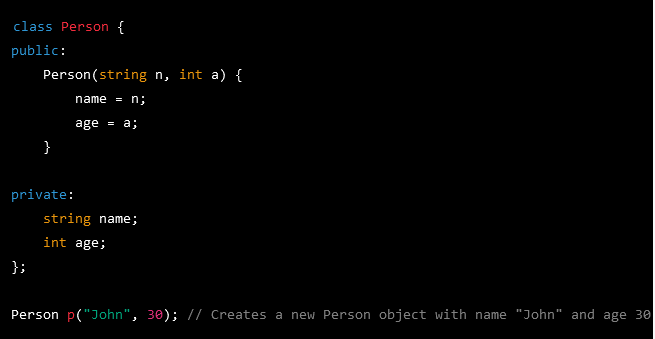
Bu erda C++ tilida konstruktor va destruktordan foydalangan holda kutubxona dasturiga misol keltirilgan. Bu dastur kitob sinfini nom, muallif va nashr yili bilan belgilaydi va yangi Book obyekti yaratilganda ushbu qiymatlarni ishga tushirish uchun konstruktordan foydalanadi. Dastur shuningdek, Book ob’ekti yo‘q qilinganda ajratilgan har qanday resurslarni chiqarish uchun destruktordan foydalanadi.

Ushbu dastur ishga tushirilganda u "Buyuk Getsbi", muallifi "F. Skott Fitsjerald" sarlavhali va 1925 yil nashr etilgan yangi Kitob ob’ektini yaratadi. Keyin u kitob ma’lumotlarini konsolga chop etadi va uni kompyuterga yozadi. "book\_info.txt" deb nomlangan fayl. Nihoyat, dastur Kitob ob’ekti yaratilganda ajratilgan har qanday resurslarni chiqarish uchun avtomatik ravishda Kitob destruktorini chaqiradi.

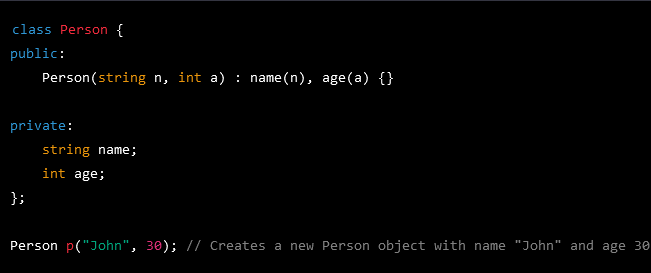
1. **A’zo o‘zgaruvchilarni initsializatsiyalash.**

C++ da sinfning aʼzo oʻzgaruvchilari konstruktorlar, ishga tushirish roʻyxatlari yoki standart aʼzo **initsializatorlari** yordamida ishga tushirilishi mumkin. Bu erda har bir yondashuvning tushuntirishi:

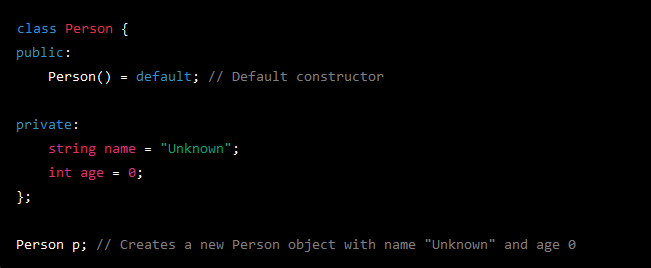
1. Konstruktorni ishga tushirish: Ushbu yondashuvda a’zo o‘zgaruvchilar sinf konstruktori yordamida ishga tushiriladi. Mana bir misol:



1. Initializer List Initialization (Initializer ro‘yxatini ishga tushirish): Bu yondashuv a’zo o‘zgaruvchilarni to‘g‘ridan-to‘g‘ri konstruktorning ishga tushirish ro‘yxatida ishga tushirish imkonini beradi. Bu konstruktor tanasida a’zo o‘zgaruvchilarni ishga tushirishdan ko‘ra samaraliroq bo‘lishi mumkin. Mana bir misol:



1. Standart a’zolarni ishga tushiruvchilar: Bu yondashuv sinf ta’rifida a’zo o‘zgaruvchilari uchun standart qiymatlarni taqdim etish imkonini beradi. Bu konstruktor yoki ishga tushirish ro‘yxatiga ehtiyojni yo‘qotib, kodni soddalashtirishi mumkin. Mana bir misol:

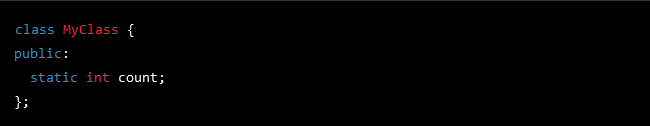


**ESLAB QOLING!!! standart a’zoni ishga tushirish usuli faqat C++ 11 va undan keyingi versiyalarida mavjud.**

1. **Sinf a’zolarining statik o‘zgaruvchilarini aniqlash va initsializatsiyalash.**

C++ da sinf a’zolarining statik o‘zgaruvchilari sinfning barcha misollari o‘rtasida taqsimlanadigan o‘zgaruvchilardir. Bu erda C++ da sinf a’zolarining statik o‘zgaruvchilari haqida bilish kerak bo‘lgan muhim narsalar mavjud:

1. Statik o‘zgaruvchilar sinf deklaratsiyasida "static" kalit so‘zi yordamida e’lon qilinadi. Masalan:



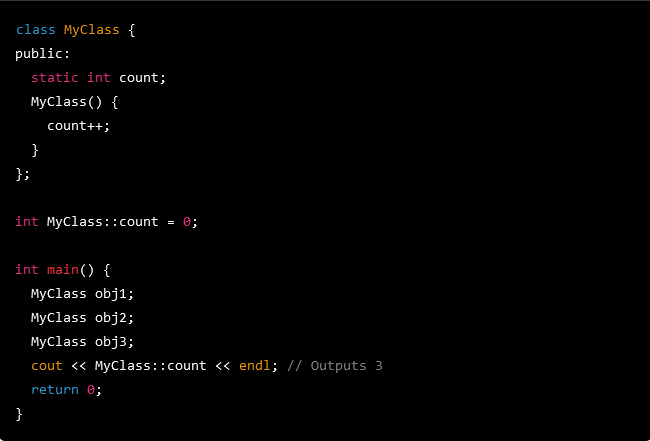
1. Statik o‘zgaruvchilar sukut bo‘yicha nolga tenglashtiriladi.

2. Statik o‘zgaruvchilar sinfning biron bir alohida ob’ektiga bog‘lanmagan. Buning o‘rniga ular sinfning barcha ob’ektlari orasida taqsimlanadi.

3. Statik o‘zgaruvchilarga kirish doirasini aniqlash operatori :: va sinf nomidan foydalanish mumkin. Masalan, MyClass sinfining statik o‘zgaruvchilari soniga kirish uchun siz MyClass::count dan foydalanishingiz mumkin.

4. Statik o‘zgaruvchilarni o‘zgartirish va sinfdan ham, sinf ob’ektlaridan ham kirish mumkin.

2. Bu yerda sinfda statik o‘zgaruvchilardan foydalanishga misol:



Yuqoridagi misolda har safar MyClass sinfining yangi ob‘ekti yaratilganda statik o‘zgaruvchilar soni ortadi. Dasturning chiqishi 3 ga teng, bu MyClass sinfining uchta ob’ekti yaratilganligini ko‘rsatadi. E’tibor bering, statik o‘zgaruvchilar soniga kirish doirani aniqlash operatori :: va sinf nomi yordamida amalga oshiriladi.

C++ da sinf aʼzolarining statik oʻzgaruvchilari sinf taʼrifidan tashqarida aniqlanadi va sintaksis yordamida ishga tushiriladi:



Bu yerda <ma’lumotlar turi> - statik o‘zgaruvchining ma’lumotlar turi, <sinf\_nomi> - sinf nomi, <statik\_o‘zgaruvchi\_nomi> - statik o‘zgaruvchining nomi va <boshlang‘ich\_qiymat> - statik o‘zgaruvchiga tayinlanadigan boshlang‘ich qiymat.



Yuqoridagi misolda biz static\_var statik o‘zgaruvchisi bilan MyClass sinfini aniqlaymiz. Statik o‘zgaruvchini sinf deklaratsiyasida e’lon qilamiz va uni MyClass::static\_var = 42 sintaksisi yordamida sinfdan tashqarida aniqlaymiz. Keyin biz sinfning ikkita ob’ektini yaratamiz va statik o‘zgaruvchiga ob’ektlar va sinf nomidan foydalanamiz. Nihoyat, biz statik o‘zgaruvchining qiymatini o‘zgartiramiz va o‘zgarish sinfning barcha ob’ektlarida aks ettirilganligini tekshirish uchun unga qayta kiramiz.

Shuni ta’kidlash kerakki, statik o‘zgaruvchilar faqat bir marta ishga tushiriladi va ishga tushirish main() funksiyasi chaqirilishidan oldin sodir bo‘ladi. Shuning uchun, ish vaqti davomida statik o‘zgaruvchiga kiritilgan har qanday o‘zgarishlar sinfning barcha ob’ektlarida aks ettiriladi.

1. **Statik metodlar**

C++ da statik usullar sinfning namunasiga ehtiyoj sezmasdan chaqirilishi mumkin bo‘lgan sinf usullaridir. Ular sinf ta’rifi doirasidagi funksiya deklaratsiyasida "statik" kalit so‘z yordamida e’lon qilinadi. Mana bir misol:

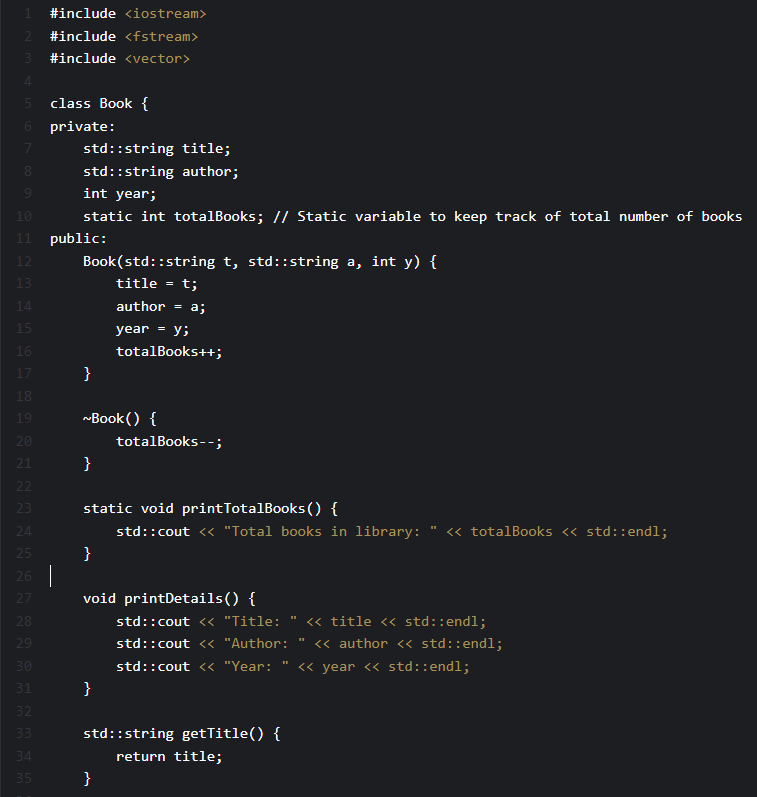


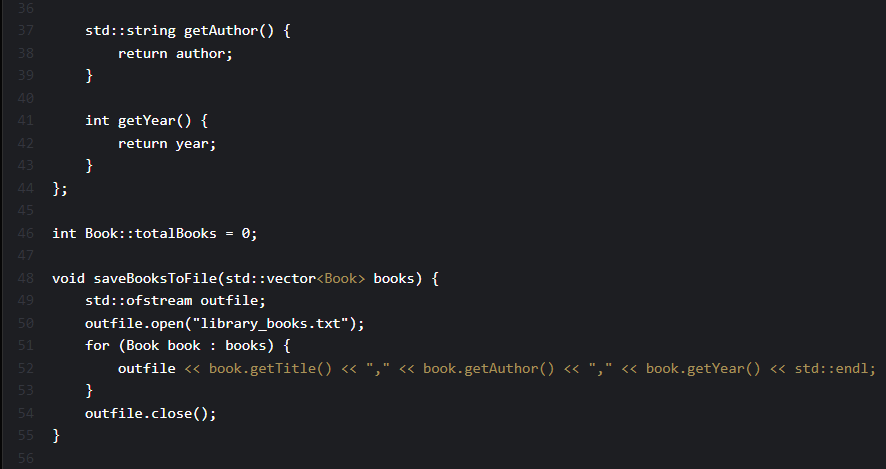
Yuqoridagi misolda biz MyClass sinfida myStaticMethod() statik usulini ishlatamiz. Biz bu usulni MyClass:: sinf nomidan so‘ng myStaticMethod() usuli nomidan foydalanib chaqirishimiz mumkin.

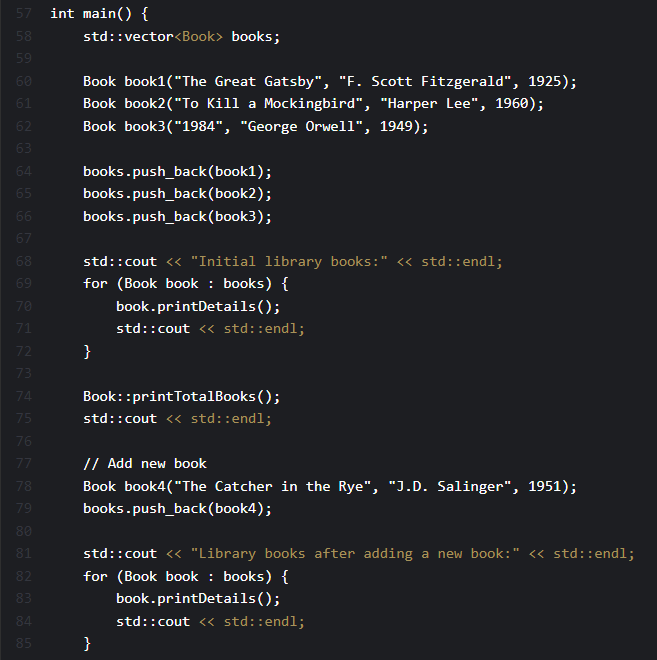
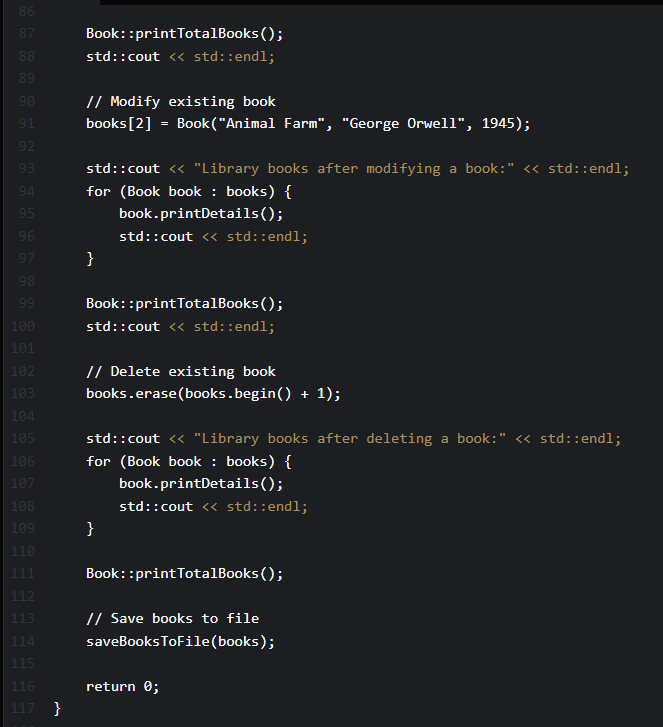
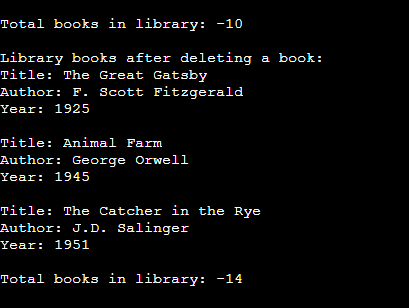
Statik usullar sinf misoliga bog‘liq bo‘lmagan operatsiyalarni bajarishimiz kerak bo‘lganda foydalidir. Masalan, matematika sinfidagi statik usul, sinf ob’ektiga ehtiyoj sezmasdan, sonning kvadrat ildizini hisoblashi mumkin.

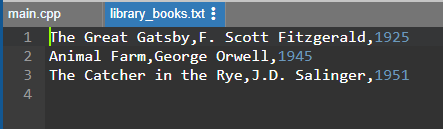
Shuni ta’kidlash kerakki, statik usulda biz statik bo‘lmagan a’zo o‘zgaruvchilari yoki sinfning usullariga kira olmaymiz. Buning sababi shundaki, statik bo‘lmagan a’zolar kirish uchun sinfning namunasini talab qiladi. Biroq, statik usullar sinfning boshqa statik a’zolariga, shu jumladan statik o‘zgaruvchilarga va boshqa statik usullarga kirishi mumkin.

**Asosiy dastur: Mana C++ tilidagi misol dasturi, u konstruktorlar, destruktorlar, aʼzo oʻzgaruvchilar, ishga tushirish va statik usullar yordamida kutubxona kitoblarini kiritish, oʻzgartirish va oʻchirish imkonini beradi. Shuningdek, u kiritilgan kitoblarni faylga saqlaydi.**







**Dastur kodlarini ==>** [**https://github.com/Samiddin/Dars-uchun**](https://github.com/Samiddin/Dars-uchun) **dan olishingiz mumkin!!!**