1-mustaqil ish.

**O’ZBEKISTON RESPUBLIKASI**

**RAQAMLI**

**TEXNOLOGIYALAR**

**VAZIRLIGI**

**MUHAMMAD AL**

**-**

**XORAZMIY NOMIDAGI TOSHKENT AXBOROT**

**TEXNOLOGIYALARI UNIVERSITETI**

**QARSHI FILIALI**

**\_\_\_\_Kompyuter injiniring\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**FAKULTETI**

**\_\_\_\_\_\_\_KI 11\_22\_\_\_\_\_\_\_\_\_\_\_\_**

**GURUH TALABASINING**

**DASTURLASH**

**2**

**FANIDAN**



**Mavzu:**

**\_\_\_\_\_Yangi nomlar fazosi uchun joy yaratish uchun\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

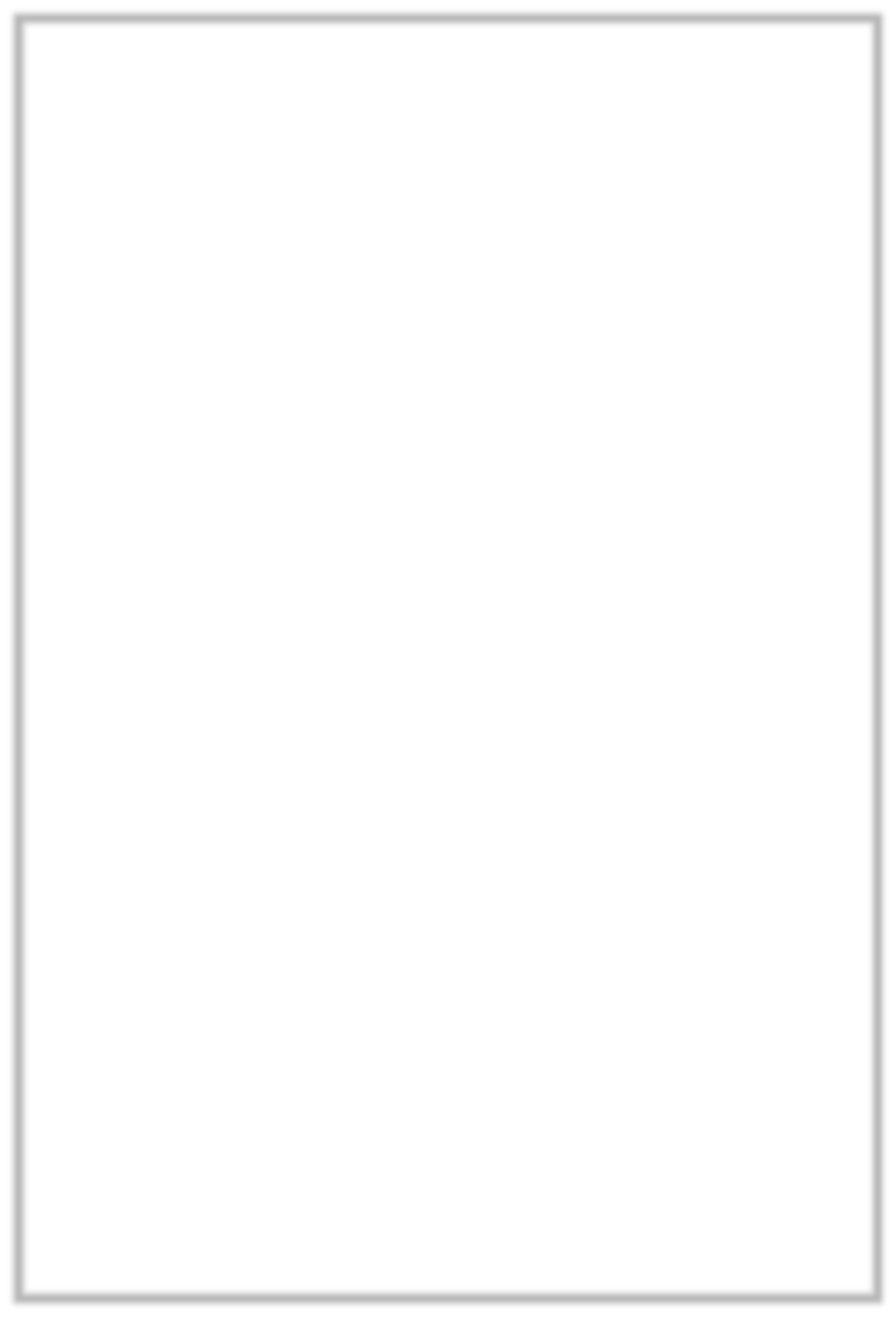
**Bajardi:**

**Mahmasharifov Shahriyor\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Qabul qildi:**

**\_\_Nosirov B\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**QARSHI**



Mavzu:Yangi nomlar fazosi uchun joy yaratish

Nomlar fazosi (namespace)-bir necha nomlar (identifikatorlar) toplamiga asosan model bo’lib, unikal indentifikatorlar (ya’ni nomlar )ni mantiqiy uchun yaratilgan abstract saqlanish joyi yoki muhitdir.

Nomlar fazosi – bu, turli xil identifikatorlar(ma’lumot tiplari nomi, funksiyalar, o’zgaruvchilar) ni aniqlovchi sohadir

Nomlar fazosi-dastur kodlarini mantiqiy guruhga birlashtirish uchun ishlatilib dasturga bir necha kutubxonalar chaqirilganda nomlar orasida o’zaro nizo kelib chiqishini oldini oladi. Nomlar fazosi barcha indentifikatorlar bir-biriga hech qanday e’lon qilinishlarsiz ishlatilishi mumkin.

Foydalanuvchi nomidan yaratilgan “Nomlar fazosi” namespace kalit so’zi orqali ko’rsatmali bloklar orqali aniqlanadi.

Namespace ism

{int bar;}

Ushbu indenfikatorlar qanday nom bilan e’lon qilingan bo’lsa shu nom bilan chaqiriladilar.

Blokdan tashqarida identifikatorlardan oldin nomlar fazosini yozish talab qilinadi.

Masalan, e’lon qilingan “namespace ism” blokdan tashqarida bar indentifikatorlari ism: : bar kabi ko’rsatilishi lozim.

#include<iostream> using namespace std; namespace ism

{ int bar;

}

Int main ()

{ ism: : bar=10; cout<<ism: : bar;

return 0 } bu kodda javobimiz 10 chiqadi. Ism: :bar shu kabi ishlatiladi.

Using direktivasi nomlar fazosidagi barcha o’zgaruvchi identifikatorlarni asosiy dasturda ishlatilishini ta’minlaydi hamda qo’shimcha prekslarni(::) ishlatilishini oldini oladi.

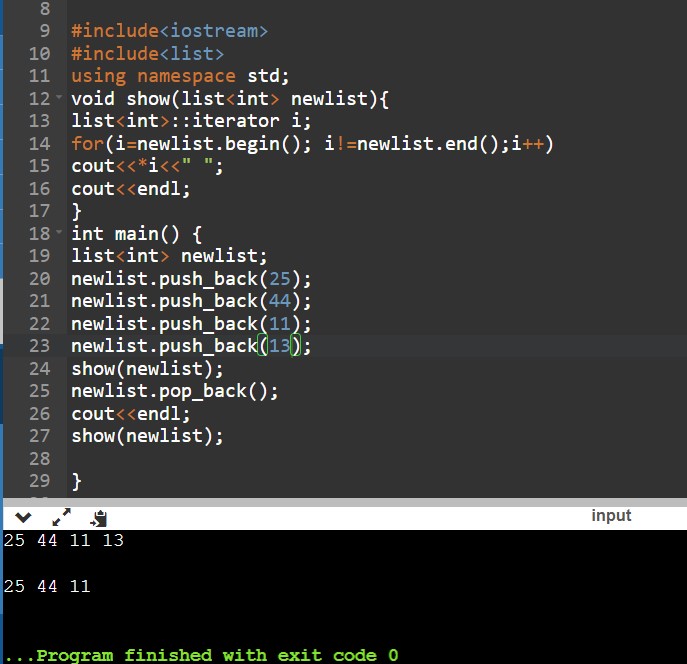
Using direktivasini nomlar fazosini aniqlagandan so’ng qo’shib qoysak ism:: prefiksi ishlatilmasa ham bo’ladi.

C++ da nomlar fazosining imkoniyati ierarxik ko’rinishga ega, ya’ni bu quyidagicha tuzilishga egadir: ovqat::sho’rva gipotetik(tahminiy) nomlar fazosi bo’lib, tovuq identifikator bo’lsin.Agar nomlar fazosi mavjud bo’lsa u ovqat::sho’rva::tovuq ko’rinishda bo’ladi.

Agar nomlar fazosi mavjud bo’masa u holda ovqat::tovuq::

ko’rsatiladi.Agar bu fazo ham mavjud bo’lmasa u holda tovuq global fazodagi identifikatorga yo’naltiriladi.

Nomlar fazosi zamonaviy dasturlashda keng qo’llanilsada, eski ko’dlarning katta qismi bunday imkoniyatga ega emas.Masalan, C++ tilining barcha standart kutubxonalari “ namespace std;”ning ichida aniqlangan, lekin (nom fazolarini ) standartlashtitrishgach ko’pgina komponetlar birlamchi global fazoda anqlanganlar.



2-mustaqil ish.

Mavzu: Konteynerlar sinflari

**Konteynerlar kutubxonasi** - bu dasturchilarga **navbat, ro'yxat va stek** kabi keng tarqalgan ma'lumotlar tuzilmalarini osonlikcha amalga oshirishga imkon beradigan sinf andozalari va algoritmlarning universal to'plamidir.

**set** – elementlar to‘plami, faqat kalitlarning qiymati bo‘yicha tartiblangan to‘plamidir, yaʻni taqqoslash amali qo‘llaniladigan, ammo takrorlanmaydigan qiymatlar — har bir kalit to‘plamda (inglizcha setto‘plam degan maʻnoni beradi) faqat bir marta foydalaniladi; Dasturga ulanish uslubi:

#include <set>

**multiset** – set, juftlikda kalitlar unikal emas, takrorlanadigan to‘plam. Agar kalit bo‘yicha qidirsangiz, siz bitta qiymatni emas, balki bir xil kalit qiymatiga ega bo‘lgan elementlar to‘plamini olasiz.

**Uch xil konteyner mavjud:**

1. **ketma-ket konteynerlar**,
2. **assosiativ konteynerlar** va
3. **tartibga solinmagan assotsiativ konteynerlar**.

Ularning har biri turli xil operatsiyalar to'plamini qo'llab-quvvatlashga mo'ljallangan.

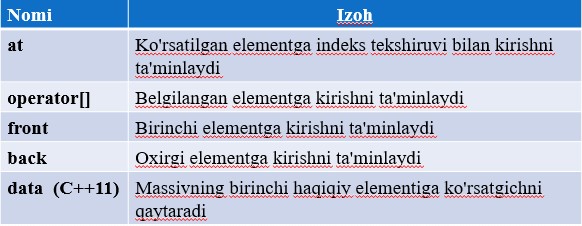
Konteyner uning elementlari uchun ajratilgan xotirani boshqaradi va ularga to'g'ridan-to'g'ri yoki iteratorlar orqali kirish uchun funksiyalarni ta'minlaydi (ko'rsatkichlarga o'xshash xususiyatlarga ega obyektlardir).

**Ketma-ket konteynerlar**

Sinf a’zolariga ketma-ket kirish huquqiga ega bo'lgan ma'lumotlar tuzilmasini amalga oshirishda ketma-ket konteynerlardan foydalaniladi.

**Ketma-ket konteynerlarni turlari:** **array -** statik doimiy massiv; **vector –** dynamik doimiy massiv; **deque -** ikki tomonlama navbat; **forward\_list –**bir tomonlama bog'langan ro'yxat; **list –** ikki tomonlama bog'langan ro'yxat.

**array ning funksiya – a’zolari**



**Iteratorlar** to’plam elementlariga kirishni ta'minlaydi. Iteratorlardan foydalanib, elementlarni takrorlash juda qulay. Iterator turi iterator tomonidan tavsiflanadi. Ammo har bir to’plam uchun iteratorning o'ziga xos turlari mavjud.  **array sinfining o’lchamlari va jarayonlari**

