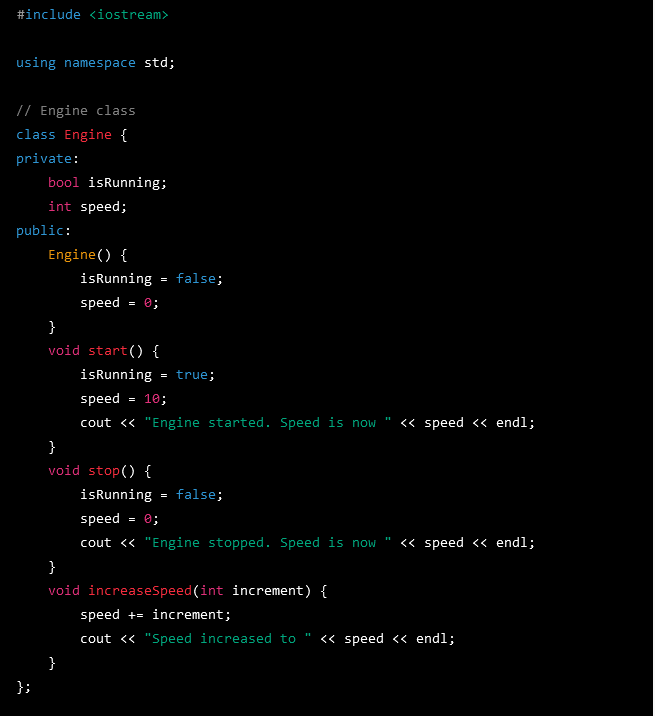
**Ma’ruza 7: Obyektlarning o‘zaro aloqasi. Bog‘lanish turlari.**

C++ dasturlash tilida ob’ektlar bir-biri bilan turli **metodlar** orqali o‘zaro bog‘lanadilar. Ob’ektlar bir birlari bilan quyidagi metodlar orqali o‘zaro bog‘lanadilar:

1. (**Method calls**) **Chaqiruv usuli**: Ob’ektlar bir-birining usullarini chaqirish orqali o‘zaro ta’sir qilishi mumkin. Misol uchun, agar sizda “Avtomobil” sinfidagi ob’ekt va “Dvigatel” sinfidagi ob’ekt bo‘lsa, “Avtomobil” ob’ekti “Dvigatel” ob’ektida uning tezligini ishga tushirish, to‘xtatish yoki o‘zgartirish uchun usullarni chaqirishi mumkin.





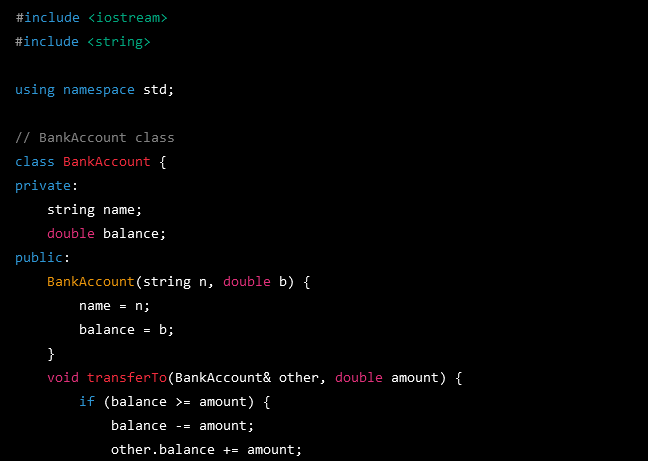
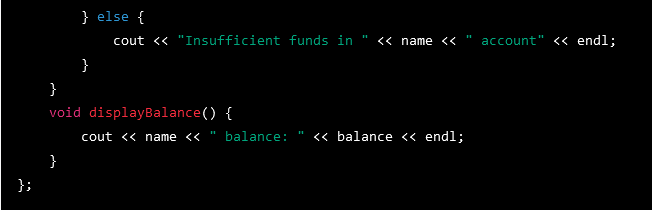
Ushbu dasturda bizda ikkita sinf mavjud: “Dvigatel” va “Avtomobil”. “Dvigatel” sinfida dvigatelni ishga tushirish va to‘xtatish, shuningdek, tezligini oshirish vosita usullari mavjud. “Avtomobil” klassi xususiy a’zo o‘zgaruvchisi sifatida “Dvigatel” ob’ektiga ega va u avtomobilni ishga tushirish, mashinani to‘xtatish va avtomobilni tezlashtirish usullariga ega.

**main()** funksiyasida biz “Avtomobil” sinfining namunasini yaratamiz va avtomobil dvigatelini ishga tushirish uchun uning **start()** usulini chaqiramiz. Keyin avtomobil tezligini oshirish uchun **accelerate()** usulini ikki marta chaqiramiz va nihoyat avtomobil dvigatelini to‘xtatish uchun **stop()** usulini chaqiramiz.

“Car” sinfining **start()** usuli chaqirilganda, u o‘z ichiga olgan “Dvigatel” ob’ektining **start()** usulini chaqiradi. Xuddi shunday, «Car» sinfining **accelerate()** usuli chaqirilganda, u avtomobil tezligini oshirish uchun «Dvigatel» ob’ektining **artırmaSpeed()** usulini chaqiradi. Nihoyat, “Car” sinfining **stop()** usuli chaqirilganda, u avtomobil dvigatelini to‘xtatish uchun “Dvigatel” obyektining **stop()** usulini chaqiradi.

Bu dastur turli sinflar ob’ektlari bir-birining usullarini chaqirish orqali bir-biri bilan qanday o‘zaro ta’sir qilishi mumkinligini ko‘rsatadi, bu bizga yanada murakkab va moslashuvchan dasturlarni yaratish imkonini beradi.

2. **Ob’ektlarni argument sifatida o‘tkazish**: Ob’ektlar boshqa ob’ektlarning usullariga argument sifatida uzatilishi mumkin. Masalan, “BankAccount” toifasidagi ob’ektda boshqa bank hisobvarag‘iga pul o‘tkazish usuli bo‘lishi mumkin va siz ikkita hisob o‘rtasida pul o‘tkazish uchun ushbu usulga argument sifatida “BankAccount” sinfining boshqa ob’ektini o‘tkazishingiz mumkin.

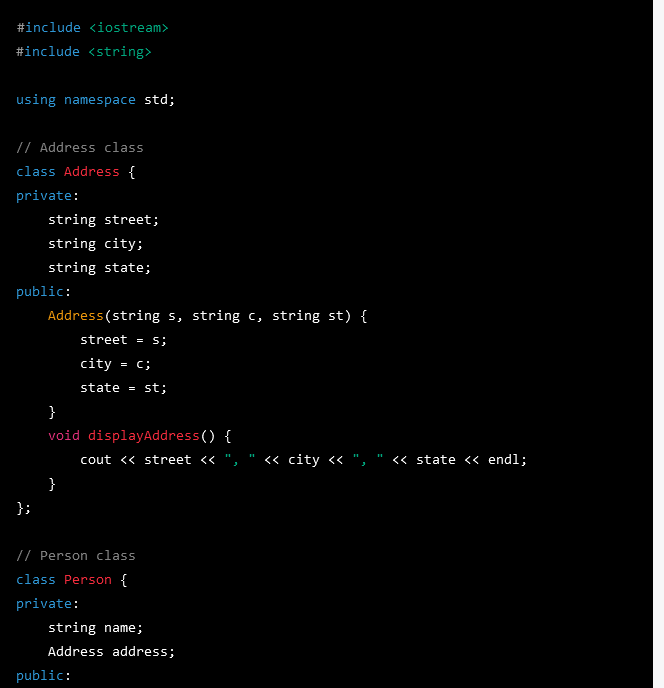
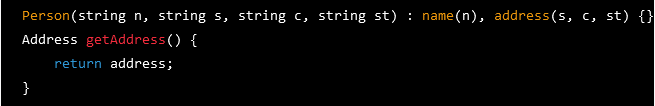
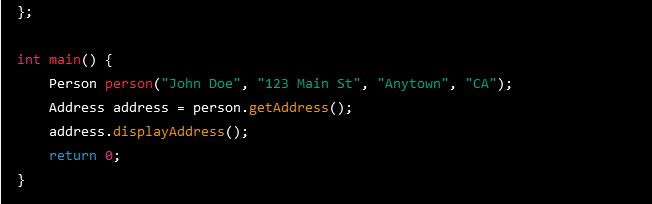
Ushbu dasturda bizda boshqa bank hisobiga pul o‘tkazish uchun “**transferTo()**” usuli mavjud “BankAccount” klassi mavjud. Ushbu usul argument sifatida boshqa “BankAccount” ob’ektiga havolani, shuningdek pul o‘tkazmalari miqdorini oladi.

**main()** funksiyasida biz “BankAccount” sinfining ikkita nusxasini yaratamiz va ularning dastlabki qoldiqlarini ko‘rsatish uchun **displayBalance()** usullarini chaqiramiz. Keyin biz birinchi hisobning **transferTo()** usulini chaqiramiz, ikkinchi hisobda argument va o‘tkazish uchun summa sifatida o‘tamiz. Agar birinchi hisobda mablag‘ yetarli bo‘lsa, pul o‘tkazmasi muvaffaqiyatli bo‘ladi va ikkala hisobning qoldiqlari ham shunga mos ravishda yangilanadi. Agar birinchi hisobda mablag‘ yetarli bo‘lmasa, o‘tkazma amalga oshmaydi va xato xabari ko‘rsatiladi.

Keyin ikkala hisobning yangilangan qoldiqlarini ko‘rsatish uchun yana **displayBalance()** usullarini chaqiramiz. Nihoyat, biz ikkinchi hisobning **transferTo()** usulini chaqiramiz, birinchi hisobda argument va o‘tkazish uchun summa sifatida o‘tamiz. Ikkinchi hisobda pul o‘tkazmasini bajarish uchun etarli mablag‘ yo‘qligi sababli, o‘tkazma bajarilmaydi va xato xabari ko‘rsatiladi.

Ushbu dastur bir xil sinf ob’ektlarini boshqa ob’ektlarning usullariga argument sifatida qanday o‘tkazish mumkinligini ko‘rsatadi, bu bizga bir xil sinfning turli misollari o‘rtasida operatsiyalarni bajarishga imkon beradi.

3. **Ob’ektlarni qaytarish**: Ob’ektlar boshqa ob’ektlarning usullari(metodlari)dan ham qaytarilishi mumkin. Masalan, “Shaxs” sinfi ob’ektida “Manzil” sinfining yangi ob’ektini yaratish va uni qo‘ng‘iroq qiluvchiga qaytarish usuli bo‘lishi mumkin.

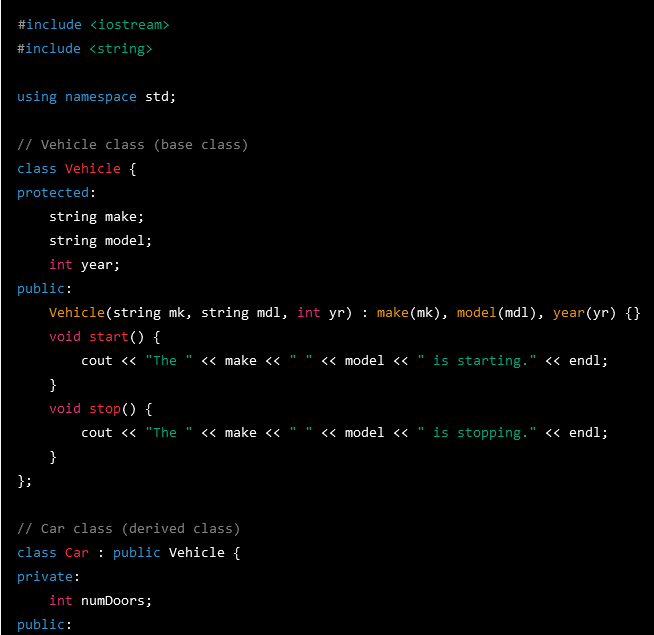
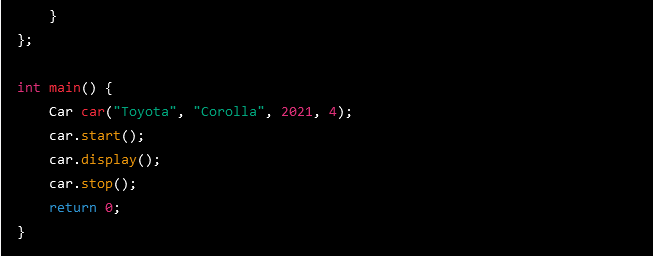
  

Ushbu dasturda bizda “Manzil” sinfining ob’ektini qaytaruvchi “**getAddress()**” usuliga ega “Shaxs” klassi mavjud. Bu usul chaqirilganda konstruktor yordamida yangi “Manzil” obyektini yaratadi va uni chaqiruvchiga qaytaradi.

**main()** funksiyasida biz “Person” sinfining yangi namunasini yaratamiz va odam manzilini olish uchun uning **getAddress()** usulini chaqiramiz. Keyin biz bu qaytarilgan ob’ektni yangi “Manzil” ob’ektiga tayinlaymiz va manzilni ko‘rsatish uchun uning **displayAddress()** usulini chaqiramiz.

Ushbu dastur boshqa ob’ektlarning usullaridan ob’ektlarni qanday qaytarish mumkinligini ko‘rsatadi, bu bizga boshqa sinflarning yangi namunalarini yaratish va ularni dasturimizda ishlatish imkonini beradi.

4. **Meros** (**Inheritance**): ob'ektlar boshqa ob'ektlardan xususiyatlar va usullarni meros qilib olishi mumkin. Misol uchun, sizda barcha turdagi transport vositalariga xos xususiyatlar va usullarni belgilaydigan «Avtomobil» deb nomlangan sinfingiz bo'lishi mumkin, so'ngra «Avtomobil» sinfidan meros bo'lgan «Avtomobil» va «Yuk mashinasi» kabi kichik sinflarni yaratib, o'ziga xosligini qo'shishingiz mumkin. xususiyatlari va usullari.

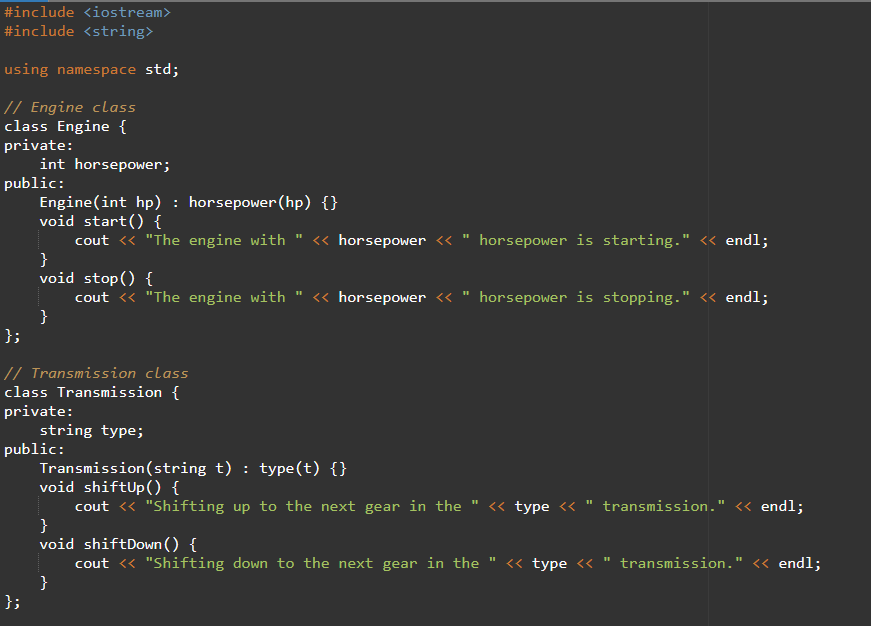
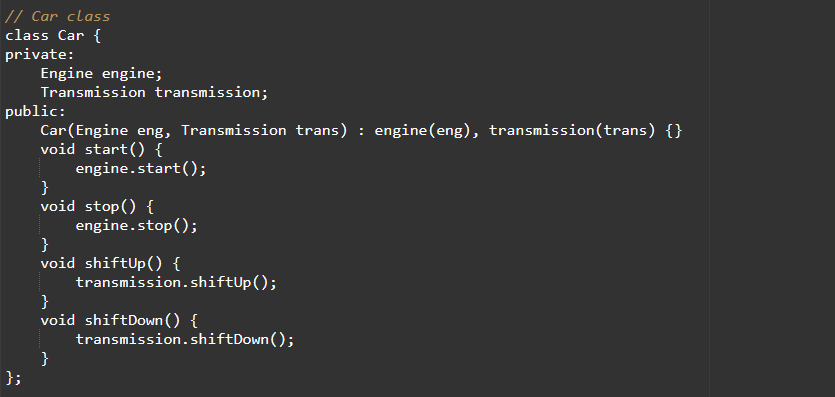
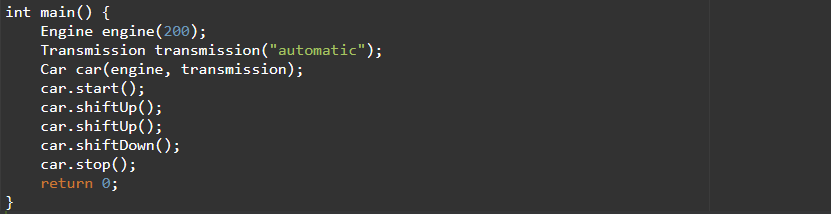
Ushbu dasturda bizda marka, model va yil kabi barcha turdagi transport vositalari uchun umumiy xususiyatlar va usullarni, shuningdek, transport vositasini ishga tushirish va to‘xtatish usullarini belgilaydigan “Avtomobil” klassi mavjud.

Keyin biz umumiy foydalanish spetsifikatsiyasi yordamida “Avtomobil” sinfidan meros bo‘ladigan “Avtomobil” sinfini yaratamiz. “Avtomobil” klassi o‘zining noyob xususiyatini, eshiklar sonini va avtomobil xususiyatlarini aks ettiruvchi **displey()** usulini qo‘shadi.

**main()** funksiyasida biz “Avtomobil” sinfining yangi namunasini yaratamiz va uning mashinani ishga tushirish, ko‘rsatish va to‘xtatish usullarini chaqiramiz. “Avtomobil” klassi “Vehicle”(Transport vositasi) sinfidan meros bo‘lganligi sababli, u avtomatik ravishda asosiy sinfda belgilangan **start()** va **stop()** usullariga kirish huquqiga ega.

Bu dastur C++ da meros qanday ishlashini va quyi sinflar yangi, aniqroq sinflarni yaratish uchun asosiy sinfga oʻz xossalari va usullarini qanday qoʻshishi mumkinligini koʻrsatadi.

5. **Tarkib (Composition)**: Ob’ektlar boshqa ob’ektlardan iborat bo‘lishi mumkin. Masalan, “Avtomobil” ob‘ekti “Dvigatel”, “Transmissiya”, “G‘ildirak” va “Tormoz” ob’ektlaridan iborat bo‘lishi mumkin, ularning har biri avtomobilning umumiy funksionalligiga hissa qo‘shadigan o‘z xususiyatlari va usullariga ega.

Ushbu dasturda bizda uchta sinf mavjud: “Dvigatel”, “Transmissiya” va “Avtomobil”. «Dvigatel» va «Transmissiya» sinflari avtomobilning alohida komponentlarini ifodalaydi va o‘z xususiyatlari va usullariga ega. “Avtomobil” sinfi “Dvigatel” va “Transmissiya” obyektidan iborat bo‘lib, ular “Avtomobil” sinfining konstruktoriga beriladi.

«Avtomobil» klassi avtomobilni ishga tushirish va to‘xtatishning o‘ziga xos usullariga, shuningdek, transmissiyada yuqoriga va pastga siljish usullariga ega. Bu usullar oddiygina «Avtomobil» ob’ektining bir qismi bo‘lgan «Dvigatel» va «Transmissiya» ob‘ektlarining mos keladigan usullarini chaqiradi.

**main()** funksiyasida biz 200 ot kuchiga ega yangi «Dvigatel» ob’ektini va «avtomatik» turdagi «Transmission» ob’ektini yaratamiz. Keyin biz yangi «Avtomobil» ob’ektini yaratamiz va «Dvigatel» va «Transmissiya» ob’ektlariga o‘tamiz. Dvigatelni ishga tushirish, uzatishda yuqoriga va pastga siljish va dvigatelni to‘xtatish uchun biz «Avtomobil» ob’ektida turli usullarni chaqiramiz.

Ushbu dastur kompozitsiyaning C++ da qanday ishlashini va qanday qilib murakkabroq tizimlarni qurish uchun ob’ektlar boshqa ob’ektlardan tuzilishi mumkinligini ko‘rsatadi.

Bular C++ da ob’ektlarning bir-biri bilan o‘zaro ta’siriga bir nechta misollardir. Muayyan o‘zaro ta’sirlar sizning dasturingizdagi ob’ektlarning dizayni va ularning bir-biriga bo‘lgan munosabatlariga bog‘liq bo‘ladi.

**Savollar:**

1. Ob’ektga yo‘naltirilgan dasturlash nima va u nima uchun foydali?
2. C++ tilidagi ob'ektlar orasidagi bog'lanishning besh turi qanday?
3. Usul chaqiruvlari orqali ob'ektlar bir-biri bilan qanday o'zaro ta'sir qilishi mumkin?
4. Meros nima va u C++ da qanday ishlaydi?
5. Kichikroq qismlardan murakkab ob'ektlarni qurish uchun kompozitsiyadan qanday foydalanish mumkin?

**Mashqlar:**

1. Har birining o‘z nomi va yoshi bo‘lgan “Shaxs” sinfining ikkita ob’ektini yaratadigan C++ dasturini yozing. Bir kishi ikkinchisiga “Salom” deyishi uchun usul chaqiruvidan foydalaning.
2. Kenglik va balandlik xususiyatlariga ega “To‘rtburchak” deb nomlangan C++ sinfini yarating. To‘rtburchakning maydonini hisoblaydigan usulni aniqlang. Ushbu sinf ob’ektini yarating va uni maydonni konsolga chop etadigan boshqa ob’ektning usuliga argument sifatida o‘tkazing.
3. Marka va model xususiyatlariga ega “Avtomobil” deb nomlangan sinf yarating. “Avtomobil” dan meros bo‘lib, eshiklar soni va yoqilg‘i turi uchun xususiyatlarni qo‘shadigan “Avtomobil” deb nomlangan kichik sinfni aniqlang. “Avtomobil” sinfining ob’ektini yarating va uning xususiyatlarini konsolga chop eting.
4. “Hayvon” sinfining yangi obэektini qaytaradigan usul bilan “Hayvon” deb nomlangan sinfni yarating. Dasturingizning asosiy funksiyasida “Hayvon” sinfining obyektini yarating va yangi obyekt yaratish uchun ushbu usulni chaqiring, so‘ngra uning xususiyatlarini konsolga chop eting.
5. "Song"(Qo‘shiq) sinfining bir nechta ob‘ektlaridan tashkil topgan “Pleylist" deb nomlangan sinf yarating. Pleylistga qo'shiqlar qo'shish va o'chirish va ro'yxatdagi keyingi qo'shiqni ijro etish imkonini beruvchi usullarni belgilang. "Pleylist" sinfining ob'ektini yarating va unga bir nechta qo'shiqlarni qo'shing, so'ngra pleylistni tinglash uchun ijro etish usulini chaqiring.