

3-Ma'ruza. C++ da o'zgaruvchilar

Reja:

- 3.1 O'zgaruvchilar
- 3.2 Konstantalar
- 3.3 Turlar bilan ishlash

Kalit so'zlar: *bad(), iostream, kiritish amali, buffer, istream, chiqarish amali, clear(), ofstream, kiritish tugallanganligi belgisi, close(), open(), eof() oqimining holati, ostream, kiritish qurilmasi, fail(), unget(), chiqarish qurilmasi, good(), qurilma drayveri, fayl, ifstream.*

3.1. O'zgaruvchilar

O'zgaruvchilar ob'ekt sifatida. C tilining asosiy tushunchalaridan biri nomlangan xotira qismi – ob'ekt tushunchasidir. Obektning xususiy xoli bu o'zgaruvchidir. O'zgaruvchiga qiymat berilganda unga ajratilgan xotira qismiga shu qiymat kodi yoziladi. O'zgaruvchi qiymatiga nomi orqali murojaat qilish mumkin, xotira qismiga esa faqat adresi orqali murojaat qilinadi. O'zgaruvchi nomi bu erkin kiritiladigan identifikatordir. O'zgaruvchi nomi sifatida xizmatchi so'zlarni ishlatish mumkin emas.

O'zgaruvchilarni ta'riflash. C tilida o'zgaruvchini aniqlash uchun kompyuterga uning turi (masalan, int, char yoki float) hamda ismi haqida ma'lumot beriladi. Bu axborot asosida kompilyatorga o'zgaruvchi uchun qancha joy ajratish lozim va bu o'zgaruvchida qanday turdagi qiymat saqlanishi mumkinligi haqida ma'lumot aniq bo'ladi. O'zgaruvchi nomi identifikator bo'lib, xizmatchi so'zlardan farqli bo'lishi kerak.

Har bir yacheyka bir bayt o'lchovga ega. Agar o'zgaruvchi uchun ko'rsatilgan tur 4 baytni talab qilsa, uning uchun to'rtta yacheyka ajratiladi. Aynan o'zgaruvchini turiga muvofiq ravishda kompilyator bu o'zgaruvchi uchun qancha joy ajratish kerakligini aniqlaydi.

Kompyuterda qiymatlarni ifodalash uchun bitlar va baytlar qo'llaniladi va xotira baytlarda hisoblanadi.

O'zgaruvchilar turlari. O'zgaruvchilarning quyidagi turlari mavjud:

char – bitta simvol;

long char – uzun simvol;

int – butun son;

short yoki **short int** – qisqa butun son;

long yoki **long int** – uzun butun son;

float haqiqiy son;

long float yoki **double** – ikkilangan haqiqiy son;

long double – uzun ikkilangan haqiqiy son.

Butun sonlar ta'riflanganda ko'rilgan turlar oldiga unsigned (ishorasiz) ta'rifi qo'shilishi mumkin. Bu ta'rif qo'shilgan butun sonlar ustida amallar mod 2^n arifmetikasiga asoslangandir. Bu yerda n soni int turi xotirada egallovchi razryadlar sonidir. Agar ishoraciz k soni uzunligi int soni razryadlar sonidan uzun bo'lsa, bu son qiymati k mod 2^n ga teng bo'ladi. Ishorasiz k son uchun $-k$ amali $2^n - k$ formula asosida hisoblanadi. Ishorali, ya'ni signed turidagi sonlarning eng katta razryadi son ishorasini ko'rsatish uchun ishlatilsa unsigned (ishorasiz) turdagi sonlarda bu razryad sonni tasvirlash uchun ishlatiladi.

O'zgaruvchilarni dasturning ixtiyoriy qismida ta'riflash yoki qayta ta'riflash mumkin.

Misol uchun:

int a, b1, ac; yoki

int a;

int b1;

int ac;

O'zgaruvchilar ta'riflanganda ularning qiymatlari aniqlanmagan bo'ladi. Lekin o'zgaruvchilarni ta'riflashda inisializasiya ya'ni boshlang'ich qiymatlarini ko'rsatish mumkin.

Misol uchun:

int i = 0;

char c = 'k';

Typedef ta'riflovchisi yangi turlarni kiritishga imkon beradi.

Misol uchun yangi COD turini kiritish:

typedef unsigned char COD;

COD simbol;

Butun turlar o'lchami. Bir xil turdagi o'zgaruvchilar uchun turli kompyuterlarda xotiradan turli hajmdagi joy ajratilishi mumkin. Lekin bitta kompyuterda bir xil turdagi ikkita o'zgaruvchi bir xil miqdorda joy egallaydi.

Masalan, char turli o'zgaruvchi bir bayt hajmni egallaydi. Ko'pgina kompyuterlarda short int (qisqa butun) turi ikki bayt, long int turi esa 4 bayt joy egallaydi. Butun qiymatlar o'lchovini kompyuter sistemasi va ishlatiladigan kompilyator aniqlaydi. 32 – razryadli kompyuterlarda butun o'zgaruvchilar 4 bayt joy egallaydi.

3.2. Konstantalar

Konstantalar turlari. Konstanta bu o'zgartirish mumkin bo'lmagan qiymatdir. C tilida besh turdagi konstantalar ishlatilishi mumkin: simvollar, butun sonlar, haqiqiy sonlar, sanovchi konstantalar va nul ko'rsatkich.

Belgili o'zgarmaslar. Belgili o'zgarmaslar odatda bir bayt joyni egallaydi va bu 256 xil belgini saqlash uchun yetarlidir. Char turi qiymatlarini 0..255 sonlar to'plamiga yoki ASCII belgilar to'plamiga interpretasiya qilish mumkin.

ASCII belgilari deganda kompyuterlarda qo'llaniladigan standart belgilar to'plami tushuniladi. ASCII - bu American Standard Code for Information Interchange (Amerikaning axborot almashinishi uchun standart kodi) degan ma'noni anglatadi.

Misol uchun 'x','*','\012','\0','\n'- bitta simvolli konstanta; 'dd','\n\t','\x07\x07' ikki simvolli konstantalar.

C kompilyatorida tekstlarni formatlovchi bir nechta maxsus belgilardan foydalaniladi. (Ulardan eng ko'p tarqalgani jadvalda keltirilgan).

Maxsus belgilar axborotlarni ekranga, faylga va boshqa chiqarish qurilmalariga chiqarishda formatlash uchun qo'llaniladi.

Maxsus '\ ' simvalidan boshlangan simvollar eskeyp simvollar deyiladi. Simvolli konstanta qiymati simvolning kompyuterda qabul qilingan sonli kodiga tengdir.

ESC (eskeyp) simvollar jadvali:

Yozilishi	Ichki kodi	Simvoli(nomi)	Ma'nosi
-----------	------------	---------------	---------

\a	0x07	bel (audible bell)	Tovush signali
\b	0x08	bs (bascpase)	Bir qadam qaytish
\f	0x0C	ff (form feed)	Sahifani o'tkazish
\n	0x0A	lf (line feed)	Qatorni o'tkazish
\r	0x0D	cr (carriage return)	Karetkani qaytarish
\t	0x09	ht (horizontal tab)	Gorizontal tabulyasiya
\v	0x0B	vt (vertical tab)	Vertikal tabulyasiya
\\	0x5C	\ (backslash)	Teskari chiziq
\'	0x27	' (single out)	Apostrif (oddiy qavs)
\"	0x22	" (double quote)	Ikkilik qavs
\?	0x3F	? (question mark)	Savol belgisi
\000	000	ixtiyoriy (octal number)	Simvol sakkizlik kodi
\xhh	0xhh	ixtiyoriy (hex number)	Simvol o'n oltilik kodi

Ma'lumotlarning butun son turi. Butun sonlar o'nlik, sakkizlik yoki o'n oltilik sanoq sistemalarida berilishi mumkin.

O'nlik sanoq sistemasida butun sonlar 0-9 raqamlari ketma-ketligidan iborat bo'lib, birinchi raqami 0 bo'lishi kerak emas. Lekin yagona 0 bo'lishi mumkin.

Sakkizlik sanoq sistemasida butun sonlar 0 bilan boshlanuvchi 0-7 raqamlaridan iborat ketma-ketlikdir.

O'n oltilik sanoq sistemasida butun son 0x yoki 0X bilan boshlanuvchi 0-9 raqamlari va a-f yoki A-F harflaridan iborat ketma-ketlikdir.

Masalan, 15 va 22 o'nlik sonlari sakkizlikda 017 va 026, o'n oltilikda 0xF va 0x16 shaklda tasvirlanadi.

Ma'lumotlarning uzun butun son turi.

Oxiriga l yoki L harflari qo'yilgan o'nlik, sakkizlik yoki o'n oltilik butun son.

Ma'lumotlarning ishorasiz (unsigned) butun son turi:

Oxiriga u yoki U harflari qo'yilgan o'nlik, sakkizlik yoki o'n oltilik oddiy yoki uzun butun son.

Ma'lumotlarning haqiqiy son turi. Ma'lumotlarning haqiqiy son turi olti qismdan iborat bo'lishi mumkin: butun qism, nuqta, kasr qism, e yoki E belgisi, o'nlik daraja va F yoki f suffikslari.

Masalan : 66., .0, .12, 3.14F, 1.12e-12.

Ma'lumotlarning uzun haqiqiy son turi:

Oxiriga L yoki l suffikslari qo'yilgan haqiqiy son.

Masalan: 2E+6L;

Sanovchi konstanta. Sanovchi konstantalar enum xizmatchi so'zi yordamida kiritilib, int turidagi sonlarga qulay so'zlarni mos qo'yish uchun ishlatiladi.

Misol uchun:

```
enum{one = 1,two = 2,three = 3};
```

Agar son qiymatlari ko'rsatilmagan bo'lsa eng chapki so'zga 0 qiymati berilib qolganlariga tartib bo'yicha o'suvchi sonlar mos qo'yiladi:

```
enum{zero,one,two};
```

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

```
zero = 0, one = 1, two = 2;
```

Konstantalar aralash ko'rinishda kiritilishi ham mumkin:

```
enum(zero,one,for = 4,five,seeks).
```

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

```
zero = 0, one = 1, for = 4;five = 5,seeks = 6;
```

Yana bir misol:

Enum BOOLEAN {NO, YES};

Konstantalar qiymatlari:

NO = 0, YES = 1;

Nul ko'rsatkich. NULL- ko'rsatkich yagona arifmetik bo'lmagan konstantadir. Konkret realizatsiyalarda null ko'rsatkich 0 yoki 0L yoki nomlangan konstanta NULL orqali tasvirlanishi mumkin. Shuni aytish lozimki, bu konstanta qiymati 0 bo'lishi yoki '0' simvoli kodiga mos kelishi shart emas.

Mantiqiy konstanta. Mantiqiy konstantalar true(rost) va false(yolg'on) qiymatlardan iborat. C tilida butun sonlar va ifodalar mantiqiy konstantalar sifatida qaraladi. Ichki ko'rinishi false – 0, ixtiyoriy boshqa qiymat true deb qaraladi.

Satrlar konstanta. Satrlar konstantalar C tili konstantalariga kirmaydi, balki leksemalari alohida turi hisoblanadi. Shuning uchun adabiyotlarda satrlar konstantalar satrlar leksemalar deb ham ataladi.

Satrlar konstanta bu ikkilik qavslarga olingan ixtiyoriy simvollar ketma-ketligidir. Misol uchun "Men satrlar konstantaman".

Satrlar orasiga eskeyp simvollar ham kirishi mumkin. Bu simvollar oldiga \ belgisi qo'yiladi. Misol uchun :

"\n Bu satr \n uch qatorga \n joylashadi".

Satrlar simvollar xotirada ketma-ket joylashtiriladi va har bir satrlar konstanta oxiriga avtomatik ravishda kompilyator tomonidan '\0' simvoli qo'shiladi. Shunday satrlar xotiradagi hajmi simvollar soni +1 baytga tengdir.

Ketma-ket kelgan va bo'shliq, tabulyasiya yoki satrlar oxiri belgisi bilan ajratilgan satrlar kompilyasiya davrida bitta satrga aylantiriladi. Misol uchun:

"Salom" "Toshkent"

satrlari bitta satrlar deb qaraladi.

"Salom Toshkent"

Bu qoidaga bir necha qatorga yozilgan satrlar ham bo'ysunadi. Misol uchun:

"O'zbekistonga "

"bahor "

"keldi"

qatorlari bitta qatorga mos:

"O'zbekistonga bahor keldi"

Agar satrda '\ ' belgisi uchrasa va bu belgidan so'ng to '\n' satr oxiri belgisigacha bo'shliq belgisi kelsa bu bo'shliq belgilari '\ ' va '\n' belgisi bilan birga satrdan o'chiriladi. Satrning o'zi keyingi satrda kelgan satr bilan qo'shiladi.

"O'zbekistonga \

bahor \

keldi"

qatorlari bitta qatorga mos:

"O'zbekistonga bahor keldi"

Nomlangan konstantalar. C tilida o'zgaruvchilardan tashqari nomlangan konstantalar kiritilishi mumkin. Bu konstantalar qiymatlarini dasturda o'zgartirish mumkin emas. Konstantalar nomlari dasturchi tomonidan kiritilgan va xizmatchi so'zlardan farqli bo'lgan identifikatorlar bo'lishi mumkin. Odatda nom sifatida katta lotin harflari va ostiga chizish belgilari kombinasiyasidan iborat identifikatorlar ishlatiladi. Nomlangan konstantalar quyidagi shaklda kiritiladi:

const tur konstanta_nomi = konstanta_qiymati.

Misol uchun:

const double EULER = 2.718282;

const long M = 999999999;

const R = 765;

Oxirgi misolda konstanta turi ko'rsatilmagan, bu konstanta int turiga tegishli deb hisoblanadi.

3.3. Turlar bilan ishlash

Turlarni keltirish. Turlarni keltirish (type casting) ma'lum turdagi o'zgaruvchi boshqa turdagi qiymat qabul qilganda foydalaniladi. Ba'zi turlar uchun keltirish avtomatik ravishda bajariladi. Avtomatik turlarni keltirish o'zgaruvchi turi hajmi qiymatni saqlashga yetarli bo'lganda bajariladi. Bu jarayon kengaytirish (*widening*) yoki yuksaltirish (*promotion*) deb ataladi, chunki, kichik razryadli tur katta razryadli turga kengaytiriladi. Bu holda turlarni avtomatik keltirish xavfsiz deb ataladi. Masalan int turi char turidagi qiymatni saqlashga yetarli, shuning uchun turlarni keltirish talab qilinmaydi. Teskari jarayon toraytirish (*narrowing*) deb ataladi, chunki qiymatni o'zgartirish talab etiladi. Bu holda turlarni avtomatik keltirish xavfli deb ataladi. Masalan haqiqiy turni butun turga keltirilganda kasr qism tashlab yuboriladi.

Amallarda turlarni avtomatik keltirish. Binar arifmetik amallar bajarilganda turlarni keltirish quyidagi qoidalar asosida amalga oshiriladi:

short va char turlari int turiga keltiriladi;

Agar operandlardan biri long turiga tegishli bo'lsa ikkinchi operand ham long turiga keltiriladi va natija ham long turiga tegishli bo'ladi;

Agar operandlardan biri float turiga tegishli bo'lsa ikkinchi operand ham float turiga keltiriladi va natija ham float turiga tegishli bo'ladi;

Agar operandlardan biri double turiga tegishli bo'lsa ikkinchi operand ham double turiga keltiriladi va natija ham double turiga tegishli bo'ladi;

Agar operandlardan biri long double turiga tegishli bo'lsa ikkinchi operand ham long double turiga keltiriladi va natija ham long double turiga tegishli bo'ladi;

Ifodalarda turlarni avtomatik keltirish. Agar ifodada short va int turidagi o'zgaruvchilar ishlatilsa, butun ifoda tiri int ga ko'tariladi. Agar ifodada biror o'zgaruvchi turi— long bo'lsa, butun ifoda turi long turga ko'tariladi. Ko'zda tutilganidek hamma butun konstantalar int turiga ega deb qaraladi. Hamma butun konstantalar oxirida L yoki 1 simvoli turgan bo'lsa, long turiga ega.

Agar ifoda float turidagi operandga ega bo'lsa, butun ifoda float turiga ko'tariladi. Agar biror operand double turiga ega bo'lsa, butun ifoda turi double turiga ko'tariladi.

Turlar bilan ishlovchi amallar. Turlarni o'zgartirish amali quyidagi ko'rinishga ega:

(tur_nomi) operand;

Bu amal operandlar qiymatini ko'rsatilgan turga keltirish uchun ishlatiladi. Operand sifatida konstanta, o'zgaruvchi yoki qavslarga olingan ifoda kelishi mumkin. Misol uchun

(long)6 amali konstanta qiymatini o'zgartirmagan holda operativ xotirada egallagan baytlar sonini oshiradi. Bu misolda konstanta turi o'zgarmagan bo'lsa, (double)6 yoki (float)6 amali konstanta ichki ko'rinishini ham o'zgartiradi. Katta butun sonlar haqiqiy turga keltirilganda sonning aniqligi yo'qolishi mumkin.

Masalan:

```
int x = 1.7+1.8;
```

```
int y = (int)1.7+(int)1.8;
```

Bu amallar bajarilishi natijasida x o'zgaruvchi qiymati 3 ga y o'zgaruvchi qiymati ikkiga teng bo'ladi.

sizeof amali operand sifatida ko'rsatilgan ob'ektning baytlarda xotiradagi hajmini hisoblash uchun ishlatiladi. Bu amalning ikki ko'rinishi mavjud:

sizeof ifoda sizeof (tur) Shuni ta'kidlab o'tish lozimki sizeof funksiyasi preprosessor qayta ishlash jarayonida bajariladi, shuning uchun dastur bajarilish jarayonida vaqt talab etmaydi.

Misol uchun:

sizeof 3.14 = 8

sizeof 3.14L = 10

sizeof 3.14f = 4

sizeof(char) = 1 sizeof(double) = 8.

Nazorat savollari

1. Zamonaviy kompyuterlarning kiritish va chiqarish vositalari qanchalik turlicha bo'lishi mumkin?
2. istream oqimi nima qiladi?
3. ostream oqimi nima qiladi?
4. Fayl nima?
5. Fayl formati nima?
6. Dasturga ma'lumotlar kiritish va chiqarish uchun ishlatiladigan 4 xil tipdagi qurilmalarni ayting..
7. Faylni o'qishning 4 bosqichini aytib bering.
8. Faylga yozishning 4 bosqichini aytib bering.
9. Oqimlarning 4 xil holatini ayting va aniqlang.
10. Ushbu kiritishga oid masalalarni yechish usullari haqida gapiring.

Foydalanilgan adabiyotlar

1. Bjarne Stroustrup. Programming: Principles and Practice Using C++ (2nd Edition). Person Education, Inc. 2014. second printing, January 2015.
2. Harry Hariom Choudhary, Bjarne M Stroustrup. C++ Programming Professional.: Sixth Best Selling Edition for Beginner's & Expert's 2014.
3. <http://www.stroustrup.com/4th.html>
4. <http://www.cplusplus.com/>