# Программчлалын C++ хэл дээр код бичин онолын асуултанд хариулах (Лаборатори №2)

Г. Саруул

ХШУИС, МКУТ-ийн Компьютерийн ухаан хөтөлбөрийн оюутан

# ОРШИЛ/УДИРТГАЛ

C++ хэл дээр нэмэгдсэн С хэл дээр байдаггүй боломжууд, дотоод мөр функц, заалтан хувьсагч зэргийг судлан онолын асуултуудыг судлаж, өгөгдсөн кодуудын үр дүнг шижилнэ.

# 2. ЗОРИЛГО

2.1 Санах ойн хаяг гэдэг ойлголт болн хаяган хувьсагчийг тодорхойлох.

2.2 Санах ойн цоорхойн талаар дэлгэрэнгүй тайлбарлах.

2.3 Хоёр хувьсагчийн утгыг солих хаяган хувьсагч болон заалтан хувьсагч параметрээрээ авдаг хэрэглэгчийн функцийг бичих. Үүнд:

2.3.1 Функцийн гол зарчимаа тодорхойлох

2.3.2 Функцүүдээр дамжуулж хаяган хувьсагч болон заалтан хувьсагч хоёрыг хэрэглэхийн давуу болон сул талыг ойлгох

# 3. ОНОЛЫН СУДАЛГАА

3.1 Санах ойн хаяг гэж юу вэ ?

Компьютерийн санах ой дээрх өгөгдлийн үүрийн дугаарыг санах ойн хаяг гэнэ.

3.2 Хаяган хувьсгач гэж юу вэ? Хаяган хувьсагчийн төрөл ямар байдаг мөн багтаамж нь хэдэн байт байдаг вэ ?

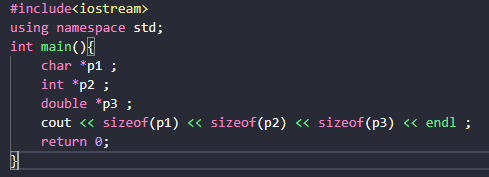
Хаяган хувьсагч гэдэг нь компьютерийн санах ой дээрх үүрийн дугаарыг /хаягийг/ хадгалдаг хувьсагч. Уг хувьсагчийн төрөл нь pointer бөгөөд хэмжээ нь өгөгдлийн төрлөөс биш, үйлдлийн системээс хамаарч өөр өөр байдаг. Жишээ нь 16 bit 2 byte, 32 bit 4 byte, 64 bit 8 byte.

3.3. Санах ойн цоорхойн талаар дэлгэрэнгүй тайлбарла

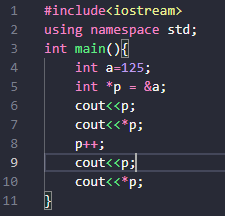
Ойн цоорхой гэдэг нь санах ой дээр ямар нэг ашиглалтгүй, дахин ашиглах боломжгүй нөөцлөөд орхисон санах ой үүсгэхийг ойн цоорхой гэнэ. Функц дотор зарласан хувьсагч тэр функцийнхаа бие дотор хүчинтэй байдаг ба тэр функц ажиллаж дуусахад тэр хувьсагч устдаг. Харин “new” түлхүүр ашиглан allocate хийгдсэн ой чөлөөлөгддөггүй. Тийм болохоор delete ашиглан өмнө нь allocate ой-г чөлөөлөх нь чухал юм. Хэрвээ тэгэхгүй бол allocate хийсэн ой ашиглагдахгүй, устахгүй үлдэнэ.

# 4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Доорх кодыг туршиж үр дүнг тайлбарлан бич.



Учир нь заагч хувьсагч санах ойд өгөгдлийн төрлөөс биш, үйлдлийн системээс хамаарч өөр зай эзэлдэг. Жишээ нь: 32 bit үйлдлийн систем дээр 4 byte зай эзэлдэг, харин 64 bit үйлдлийн системд 8 byte зай эзэлдэг.

4.2. Доорх кодыг туршиж мөр бүрийн үр дүнг тайлабарлан бич.

4-р мөр дээр int төрлийн a гэдэг нэртэй хувьсагч зарлан түүндээ 125 гэсэн утга олгож байна.

5-р мөр дээр заагч төрлийн p хувьсагч зарлан түүндээ a хувьсагчийн хаягийг оноож өгч байна.

6-р мөр дээр p хаяган хувьсагчийн хадгалж байгаа а хувьсагчийн хаягийг хэвлэж байна.

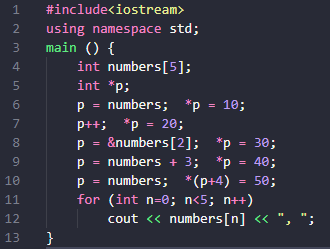
7-р мөр дээр p хаяган хувьсагчид хадгалагдаж байгаа а хувьсагчийн утгыг хаягаар нь дамжуулж хэвлэж байна. Үр дүн 125-г хэвлэнэ.

8-р мөр дээр p хувьсагч нь бүхэл тоон төрлийнх болохоор дараагийн 4 byte хаяг руу шилжүүлж зааж байна.

9-р мөр дээр 4 byte аар шилжсэн а хувьсагчийн хаяг дээр 4 byte нэмсэн хаягийг хэвэлж байна.

10-р мөр дээр p хаяган хувьсагчид байгаа а хувьсагчийн дараагийн хаяг дээр оноогдсон утгыг хэвлэж байна.

4.3. Доорх кодын мөр бүрийг тайлбарла.





4-р мөр дээр int төрлийн 5 элементтэй хүснэгт зарлав.

5-р мөр дээр хаяган хувьсагч зарлав.

6-р мөр дээр p хувьсагчид numbers хүснэгтийн хаягийг оноож өгч байна. 20 байт хэмжээтэй санах ойн эхний 4 байт-д 10 гэсэн утга хадгалж байна.

7-р мөр дээр p заагчийг дараагийн 4 байт хаяг руу шилжүүлж байна. Шилжүүлсний дараа тэр ойд 20 гэсэн утга хадгалж байна.

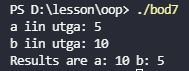
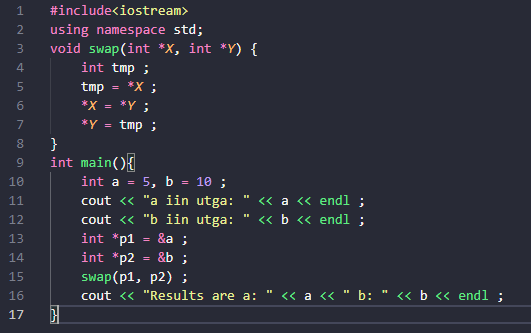
8- мөр дээр p заагчийг дараагийн 4 байт руу шилжүүлсэнтэй адилхан буюу хүснэгтийн индексээр гуравдахь элементэд 30 гэсэн утгыг хадгалж байна.

9-р мөр дээр p = numbers + 3 ; Хүснэгт нь өөрөө хаяган хувьсагч болохоор p = numbers[0] + 3 гэж ойлгож болно.

10-р мөр дээр number[0] хаягийг оноож өгч байгаад \*(p+4) үйлдлээр хамгийн сүүлийн 4 байт ийн хаягийг зааж 50 гэсэн утгаг хадгалж өгч байна.

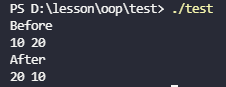
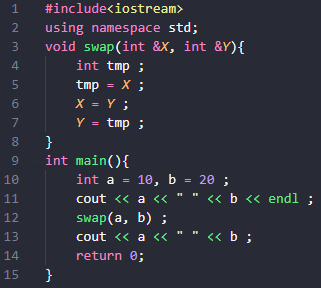
11-12-р мөр дээр for давталт ашиглан хүснэгтээр эхнээс нь төгсгөл хүртэл нь гүйлгэж уг хүснэгтийн бүх элементийг таслалын зайтай хэвлэж байна.

4.4. Хаяган хувьсагч ашиглан (функцын параметер нь хаяган хувьсагч байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич.

Swap функц нь хоёр хаяган хувьсагч авдаг ба хаяган хувьсагч ашиглан 2 хувьсагчийн утгын нь солидог. Ингэхдээ эхний хувьсагчийн утгыг хаягаар нь дамжуулж tmp хувьсагчид хадгалаад, хоёр хувьсагчийн хаягыг хооронд нь сольж, tmp хувьсагчид хадгалсан хаягаа нөгөө хаяган хувьсагчид өгч утгыг солбидог.

4.5**)**Заалтан хувьсагч (функцын параметер нь заалт байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич



Энэ функц нь хоёр хувьсагчийн утгыг заалтан хувьсагч ашиглан сольдог функц. Ингэхдээ 4.4-тэй үйл явцын хувьд адилхан. Гэхдээ заалтан хувьсагч ашиглаж хийж байгаа болохоор санах ойд илүү зай нөөцлөхгүйгээр хоёр хувьсагчийн утгыг хооронд нь сольж болно.

# ДҮГНЭЛТ

Програмд хэрэглэсэн санах ойгоо устгахгүй бол програм яваандаа санах ой хүрэлцэхгүйн улмаас зогсож санах ойн цоорхой аюултай. Заагч хувьсагч ашиглах юм бол заагч хувьсагч нь өөрөө санах ойд зай хэрэглэх болоохоор заалтан хувьсагч хэрэглэвэл заагч хувьсагчийн хэрэглэх санах ойг хэрэглэхгүйгээр илүү бага санах ой ашиглах боломжтой юм байна.

# 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.

2. Лекцийн материал

# 7. ХАВСРАЛТ

Заалтан хувьсагч ашиглаж хоёр хувьсагчийн утгыг хооронд нь солих програм

#include<iostream>

using namespace std;

void swap(int &*X*, int &*Y*){

    int tmp ;

    tmp = *X* ;

*X* = *Y*  ;

*Y* = tmp ;

}

int main(){

    int a = 10, b = 20 ;

    cout << a << " " << b << endl ;

    swap(a, b) ;

    cout << a << " " << b << endl ;

    return 0;

}

Хаяган хувьсагч ашиглаж хоёр хувьсагчийн утгыг хооронд нь солих програм

#include<iostream>

using namespace std;

void swap(int \**X*, int \**Y*){

    int tmp ;

    tmp = \**X* ;

    \**X* = \**Y* ;

    \**Y* = tmp ;

}

int main(){

    int a = 5, b = 10 ;

    cout << "a iin utga: " << a << endl ;

    cout << "b iin utga: " << b << endl ;

    int \*p1 = &a ;

    int \*p2 = &b ;

    swap(p1, p2) ;

    cout << "Results are a: " << a << "b: " << b << endl ;

}