# Програмчлалын C++ Хэл Дээр Онолын Асуултад Хариулж Мөн Код Бичих Лабораторийн Ажил (Лаборатори №2)

Ү.Мэндсайхан

МУИС, ХШУИС, Компьютерын ухаан, mendsaikhan@gurvanshiidel.mn

# 1. ОРШИЛ

Програмчлалын C++ хэл дээрх санах ойн хаяг, хаяган хувьсагч, заалтан хувьсагч, санах ойн цоорхой зэрэг санах ойтой холбоотой зүйлсийг судлан өгөгдсөн судлан ойлгож түүнийгээ ашиглан лабораторийн ажил дээр өгөгдсөн асуултад хариулж, кодуудын үр дүнг шинжилнэ.

# 2. ЗОРИЛГО

Программын санах ойн хувьсагчийн тухай ойлголттой болон, түүнийгээ ашиглан хоёр хувьсагчийн утга солих хэрэглэгчийн функцийн кодыг бичих. Санах ойн цоорхойн талаар мэдлэгтэй болох зорилготой.

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Санах ойн хаяг гэж юу вэ?

Компьютерын санах ой дээрх үүрүүд нь дугаартай байх ба санах ойн үүрийн дугаарыг санах ойн хаяг гэнэ.

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

Хаяган хувьсагч гэдэг нь санах ойн үүрийн дугаарыг хадгалдаг хувьсагч юм. Хаяган хувьсагчийн төрөл нь pointer бөгөөд багтаамж нь тухайн үйлдлийн системээс хамаарч 32 бит үйлдлийн систем бол 4 байт, 64 бит үйлдлийн систем бол 8 байт байна.

## 3.3 New оператор гэж юу вэ? ямар үйлдэл хийж юу буцаадаг вэ?

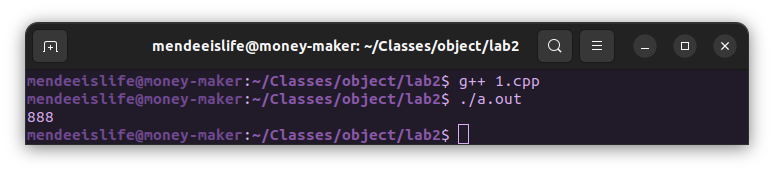
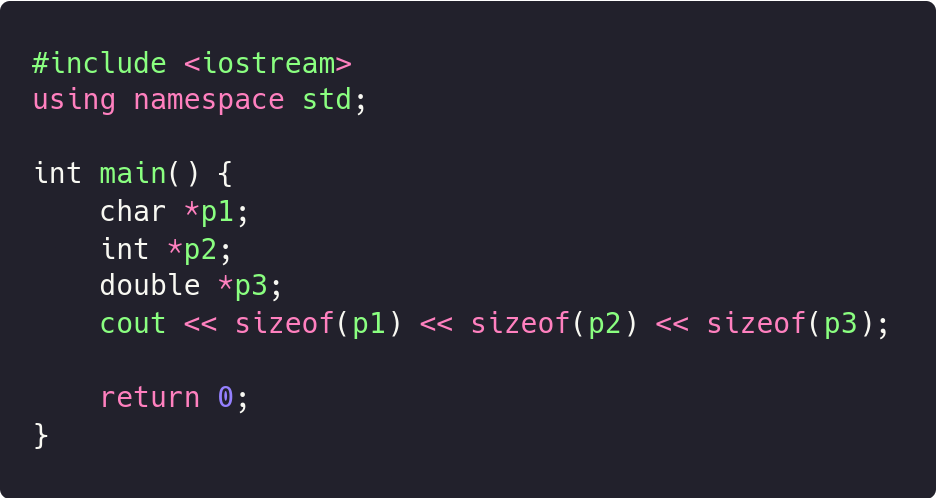
New operator гэдэг нь сул санах ойн хэсгээс санах ой нөөцлөхөд хэрэглэдэг. Сул санах ойн хэсгээс санах ой нөөцлөн тухайн программд оноож өгөөд түүнийхээ эхлэл хаягийг буцаадаг. Мөн тухайн санах ойн сул хэсгийг өөр программууд ашиглахыг хийхийг хориглодог. Хэрвээ санах ой нөөцлөх боломжгүй тохиолдолд 0 буюу null утга буцаадаг. Иймд new operator ашиглан буцаж ирсэн хаягийг ашиглахаасаа өмнө заавал шалгадаг.

## 3.4 Санах ойн цоорхойн талаар дэлгэрэнгүй тайлбарла. Ойн цоорхой гэж юу болох, ямар тохиолдолд үүсэх, яаж сэргийлэх гэх мэт.

Санах ойн цоорхой гэдэг нь санах ой дээр дахин ашиглах боломжгүй нөөцлөөд орхисон санах ой үүсгэхийг хэлнэ. Функц дотор зарласан хувьсагч тухайн функц дотроо хүчинтэй байдаг бөгөөд тухайн функц дуусахад устдаг. Харин new operator ашиглаж нөөцөлсөн санах ой чөлөөлөгддөггүй. Тэгэхээр өмнө авсан хаяган хувьсагч устахаар нөөцөлсөн санах ой руугаа хандах боломжгүй болоод ойн цоорхой үүснэ. Санах ойн цоорхойгоос сэргийлэхийн тулд хаяган хувьсагч устахаас өмнө delete оператор ашиглан нөөцөлсөн санах ойг чөлөөлөх хэрэгтэй.

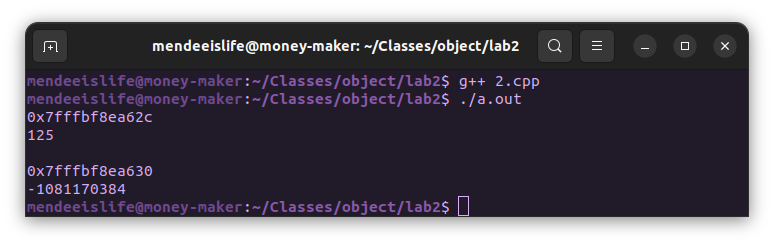
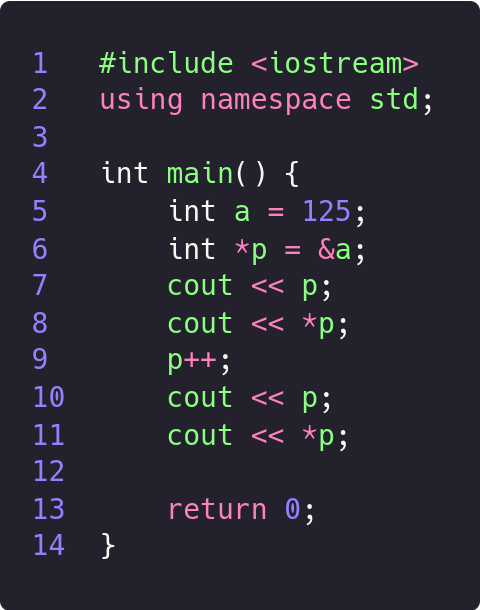
# 4. ХЭРЭГЖҮҮЛЭЛТ БОЛОН ҮР ДҮН

## 4.1 Доорх кодыг туршиж үр дүнг тайлабарлан бич.



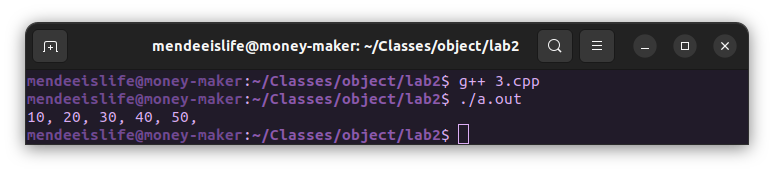
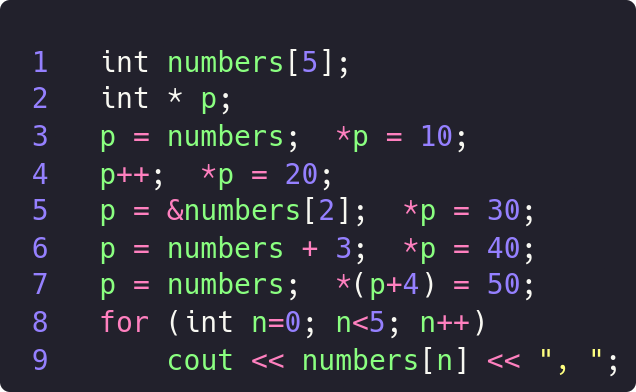
Дээрх кодыг туршиж үзэн үр дүн нь 8 8 8 гарсан. Учир нь заагч хувьсагч нь өгөгдлийн төрлөөс хамаарахгүйгээр үйлдлийн системээс хамаарч өөр өөр зай санах ой дээр эзэлдэг. Туршиж үзсэн систем нь 64 бит байсан учир хэмжээ нь 8 байт гарсан.

## 4.2 Доорх кодыг туршиж мөр бүрийн үр дүнг тайлбарлан бич.



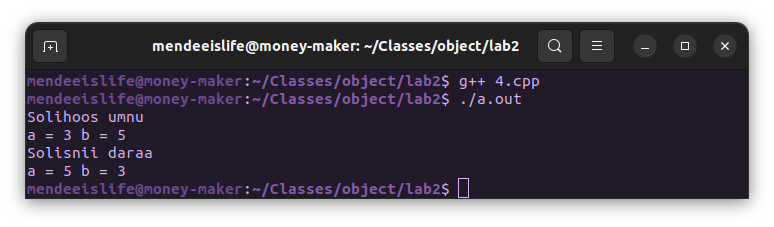
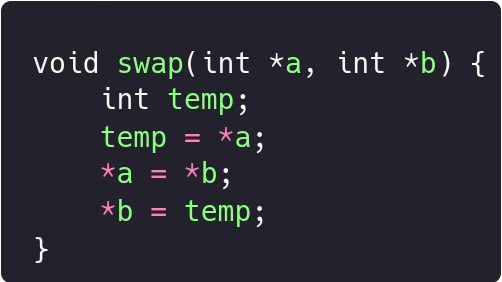
5-р мөр дээр бүхэл тоон төрлийн a хувьсагч зарлаад түүндээ 125 гэсэн утга өгч байна.  
6-р мөр дээр заагч төрлийн p хувьсагч зарлан түүндээ a хувьсагчийн хаягийг өгч байна.  
7-р мөр дээр p хаяган хувьсагч буюу a хувьсагчийн хаягийг хэвлэж байна.  
8-р мөр дээр p хаяган хувьсагчид хадгалагдаж байгаа a хувьсагчийн утгийг хаягаар нь дамжуулж хэвлэж байна. Үр дүнд 125 гэж хэвлэнэ.  
9-р мөр дээр p хаяган хувьсагчийн утгийг нэгээр нэмэгдүүлж байгаа нь p-г дараагийн 4 байт хаяг руу шилжүүлж зааж байна.   
10-р мөр дээр 4 байт аар шилжсэн p хаягийг хэвлэж байна.  
11-р мөр p хаяган хувьсагчид байгаа a хувьсагчийн дараагийн хаяг дээр оноогдсон утгыг хэвлэж байна.

## 4.3 Доорх кодыг туршиж мөр бүрийн үр дүнг тайлбарлан бич.



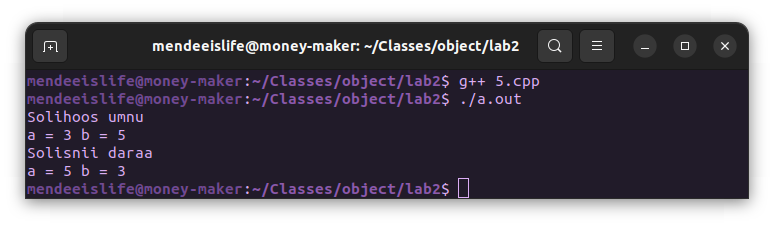
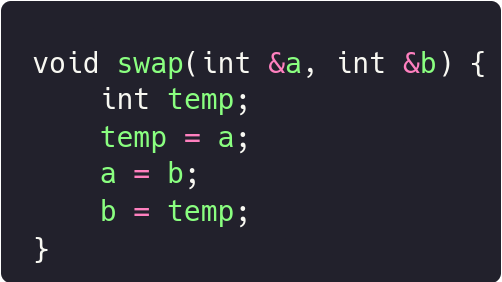
1-р мөр дээр бүхэл тоон төрлийн 5 элементтэй хүснэгт зарлав.  
2-р мөр дээр p хаяган хувьсагч зарлав.  
3-р мөр дээр хувьсагчид numbers хүснэгтийн хаягийг оноож өгөөд 20 (4 \* 5) байт санах ойн эхний 4 байт-д 10 гэсэн утга өгсөн.  
4-р мөр дээр p заагчийг нэгээр нэмэгдүүлэн дараагийн хаяг буюу дараачийн 4 байт хаяг руу зааж өгөв. Тухайн зааж өгсөн ойд 20 гэсэн утга өгсөн.  
5-р мөр дээр p заагчийг numbers хүснэгтийн 3 дахь элементийн хаягруу заагаад тэр заасан ойд 30 гэсэн утга өгсөн.  
6-р мөр дээр number + 3 гэсэн нь number нь өөрөө хаяган хувьсагч болхоор тэр дээр 3 ийг нэмхээр эхнээсээ 3 \* 4 = 12 байт хаяг руу шилжээд түүндээ 40 гэсэн утга өгсөн.  
7-р мөр дээр p хувьсагчид хүснэгтийн эхний хаягийг зааж өгөөд \*(p+4) = 50 үйлдэл нь хүснэгтийн 4 дахь элементийн хаягруу 50 гэсэн утга өгөв. P = numbers; \*(p+4) = numbers[4]; гэж ойлгож болно.  
8 болон 9-р мөр дээр нөхцөлт давталт ашиглан numbers хүснэгтийн эхнээс нь төгсгөл хүртэл давтаад тухайн элементийг ард нь таслалтай 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 temp буюу түр хадгалах хувьсагчид хадгалаад, дараа нь тухайн эхний хувьсагчийн утгыг нөгөө хувьсагчийн утгаар сольж, temp хувьсагчид хадгалсан утгаа нөгөө хаяган хувьсагчид өгч утгыг нь сольсон.

## 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. Ажиллах зарчимын хувьд 4.4-тэй адилхан. Гэхдээ заалтан хувьсагч ашиглаж хийж байгаа болохоор санах ойд илүү зай нөөцлөхгүйгээр хоёр хувьсагчийн утгыг хооронд нь сольж болно.

5. ДҮГНЭЛТ

Санах ойн талаар мэдлэгтэй болж авснаар программд хэрэглэсэн санах ойгоо чөлөөлж өгөхгүй бол санах ойн цоорхой үүсч улмаар санах ой дүүрч программ ажиллахгүй гацах, систем удаашрах зэрэг асуудлууд үүсэхээс сэргийлэх. Заалтан хувьсагч ашигласнаар хаяган хувьсагч нэмэлтээр ашиглахгүйгээр санах ой хэмнэх боломжтой боллоо.

# 6. АШИГЛАСАН МАТЕРИАЛ

1. Pointer, <https://cplusplus.com/doc/tutorial/pointers/>  
2. New Operator, [https://cplusplus.com/reference/new/operator%20new/](https://cplusplus.com/reference/new/operator new/)

# 7. ХАВСРАЛТ

Github линк, [Lab1](https://github.com/mendeeislife/object/tree/main/lab1)