# Санах ойн хаяг, хаяган хувьсагч, заалт ашиглах лабораторийн ажил (Лаборатори №2)

М.Солонго

ХШУИС, мэдээллийн систем, 20b1num2104@stud.num.edu.mn

# 1. ОРШИЛ

Хэрэглэгчийн функц, хаяган хувьсагч, заагч ашиглан хоёр хувьсагчийн байрыг солин хүснэгтэд хаягаар хандаж байгаа кодыг тайлж уншин тайлбар бичилт, дүн шинжилгээ хийж санах ойн хаяг, new operator, ойн цоорхойн тодорхойлолтыг объект програмчлалын номноос судлан дүгнэлт бичиж, хавсралт кодыг хавсаргав.

# 2. ЗОРИЛГО

Iostream сан, толгой файлыг ашиглан эхний лабораторийн хүрээнд C хэлний мэдлэг сэргээх болон cin cout хэрэглэсэн ба даалгаврыг хийхдээ дараах зорилтуудыг тавьж ажилласан:

1. Лабораторын ажлын онолын мэдлэг шалгах дасгал даалгаврын дагуу объект хандлагат технологийн С++ програмчлалын номыг уншиж судлах
2. Кодыг ажиллуулан үр дүнг хэвлэх, дүгнэлт шинжилгээ хийх
3. Хаяган хувьсагч ашиглан хэрэглэгчийн функц тодорхойлох
4. Заалтан хувьсагч ашиглан хэрэглэгчийн функц тодорхойлох

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 санах ойн хаяг

Санах ойн үүрийн дугаарыг хэлэх ба хаяган хувьсагч хадгалдаг хаягыг санах ойн хаяг гэнэ.

## 3.2 хаяган хувьсагч, хаяган хувьсагчийн төрөл, багтаамж

Хаяган хувьсагч гэж хаяг хадгалдаг хувьсагчийг хэлнэ. Зарлахын тулд урдаа \* бичнэ.

## 3.3 new operator, ямар үйлдэл хийж буцаадаг

new operator нь санах ой нөөцөлж эхлэлийн хаягийг буцаадаг. Уг операторыг голдуу хаяган хувьсагчтай холбож хэрэглэдэг.

# 3.4 санах ойн цоорхой, ямар тохиолдолд үүсэх, хэрхэн устгах

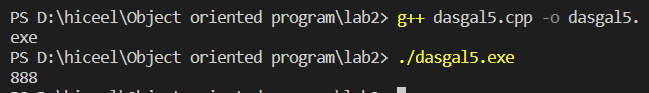
Санах ойн цоорхой нь new operator-оор үүсгэгдсэн санах ойг дахин ашиглах боломжгүй болсон тохиолдолд үүснэ. Өөрөөр хэлбэл уг үүсгэгдсэн санах ойн эхний байрлалыг хадгалж байсан хувьсагч устах үед үүснэ. Үүнээс сэргийлэхийн тулд delete operator ашиглана. Auto хувьсагч нь үйлчлэх хүрээ нь дуусах үед устах тул устахаас өмнө буюу } хаалтны өмнө delete operator-ыг бичиж өгнө.

# 4. ХЭРЭГЖҮҮЛЭЛТ

1. Объект програмчлалын номыг асуултын хүрээнд дэлгэрүүлэн уншиж судалсан.

2. кодыг visual studio code програм дээр туршин хариуг тайлбарлан бичив

3. хэрэглэгчийн функцыг даалгаврын дагуу тодорхойлох хэрэгтэй бол тодорхойлж шаардлагагүй үед нь main функц дээр шууд код бичиж даалгаврыг гүйцэтгэсэн.

Дасгал 5:

Operating system 64 bit учир хаяган хувьсагчийн size 8 гэж хэвлэсэн. Хэрэв 32 bit байсан бол 4 byte гэж хэвлэгдэх байсан.

# 5. ДҮГНЭЛТ

Заалтан хувьсагч дахин санах ой нөөцлөх шаардлагагүйгээр шууд санах ойд хандаж болдогоороо их давуу талтай. Ашиглахын тулд заавал анхны утга онооно. Заалтыг ашигласнаар ерөнхий нэг ойн өөр өөр нэр үүсгэж болохоос гадна алыг нь ч дуудаж хэрэглэж болно.

# 6. АШИГЛАСАН МАТЕРИАЛ

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

# 7. ХАВСРАЛТ

Дасгал 5:

#include<iostream>

using namespace std;

main(){

    int a=125;          //automat huvisagcid 125 gsn anhnii too oruulj bn

    int \*p=&a;          //int toroltei haygan huvisagcid a huvisagcin haygig hadgalj bn

    cout<<p<<endl;      //a iin haygig hadgalsan ucir haygig hevlej haruulna

    cout<<\*p<<endl;     //ug haygan deer ymr too bgag hevlen haruulna

    p++;                //int toroltei hayg ni 4byte bdg. haygiig nemegduulehed 4 byte aar nemegdene

    cout<<p<<endl;      //4 byte aar nemegdsen p iin haygig hevlen haruulna

    cout<<\*p;           //nemegdsen haygan deer ymr too bgag helen haruulna

}

Дасгал 6:

#include<iostream>

using namespace std;

main(){

       int numbers[5];          //5 хэмжээстэй хүснэгт зарласан

       int \*p;                  //int төрөлтэй хаяган хувьсагч зарласан

       p=numbers;    \*p=10;     //p хаяган хувьсагчид numbers хүснэгтийн эхний хаягийг хадгалж байна. хүснэгтийн эхний байранд 10 оруулж өгсөн

       p++;         \*p=20;      //p хаягийн дараагийн байрлал буюу 4 byte нэмэгдсэн хаягт 20 утга оруулж өгсөн

       p=numbers+3;  \*p=40;     //хүснэгтийн хаягийг 3аар нэмэгдүүлж 40 гэсэн утга оруулсан. 3аар нэмэгдэнэ гэдэг нь 12byte аар нэмэгдэнэ

       p=numbers;   \*(p+4)=50;  //хүснэгтийн хаягийг 4өөр нэмэгдүүлж 50 гэсэн утга оруулсан.

       for(int n=0;n<5;n++)     //for давталт гүйлгэх

        cout<<numbers[n]<<",";  //хүснэгтийн бүх элементийг байрлалаар нь дамжуулан хэвлэж байна.

       }

Дасгал 7:

#include<iostream>

using namespace std;

void swap(int \*a,int \*b){   //swap нэртэй утга буцаахгүй функц зарлаж байна. параметрын хувьд адил int төрөл байна

    int tmp;                //tmp нэртэй int төрөлтэй хувьсагч зарлаж өгнө. 2 хувьсагчийн байрыг солиход хэрэгтэй

    tmp=\*a;                 //tmp буюу хоосон хувьсагчид a хаяган хувьсагчийн хадгална

    \*a=\*b;                  // a хаяган хувьсагчид b хаяган хувьсагчийг хадгалж байна

    \*b=tmp;                 // b хаяган хувьсагчид а-гийн утга хадгалагдаж буй tmp-ийг хадгалж байна

}

main(){

    int \*p,a=10,b=20;       //int төрөлтэй p хаяган хувьсагч, a=10 b=20 анхны утгатай хувьсагч зарлаж өгнө

    swap(a,b);              //swap функцыг дуудаж байна.

    cout<<a<<b;             // функц ажиллаад a b хоёр тооны байр солигдсон эсэхийг шалган хэвлэж байна

    }

Дасгал 8:

#include<iostream>

using namespace std;

void exchange(int &a,int &b){   //exchange нэртэй утга буцаадаггүй int төрлийн a b заалтан хувьсагч параметртэй функц зарлаж байна

    int tmp;                    //int төрлийн tmp хувьсагч зарлаж байна. 2 хувьсагчийн байрыг солиход ашиглана

    tmp=a;                      //tmp-д а хадгалж байна

    a=b;                        //a=b гэснээр а хувьсагч b тоог заана

    b=tmp;                      //b заалтан хувьсагч tmp-ын хадгалсан тоог заана

}

int main(){

    int a=10,b=20;              //int төрөлтэй а=10 b=20 анхны утгатай хувьсагч зарлаж байна

    exchange(a,b);              //функцыг дуудаж ажиллуулж байна

    cout<<a<<b;                 //хэвлэж байна

}