

## (Лаборатори №2)

Компьютерийн ухааны хөтөлбөр

Г.Мөнх-Оргил 18b1num2048

### 1. ОРШИЛ/УДИРТГАЛ

Програм ажиллах болгонд санах ой ачааллагдаж байдаг. Тэдгээр ачааллагдаж байгаа санах ойтой хэрхэн ажиллах, ашиглах талаар суралцах.

### 2. ЗОРИЛГО

Санах ойн ойлголтуудыг сэргээн санах, түүнтэй ажиллах, хувьсагчийн хэрэглээг ойлгох, судлах, заалтан хувьсагчийн талаар дэлгэрүүлж судлах, санах ойн цоорхой гэж юу болох, түүний хөр хөнөөлийн талаар судлах.

1. Санах ой, санах ойн хаяг, түүний үүрэг тухай онолын судалгаа хийх.
2. Заагч болон заалтан хувьсагчийг функцэд дамжуулах.
3. Хаягаар хандаж санах ой дээр байрлах утгуудыг өөрчилөх.
4. Санах ойн цоорхойн талаар судлах

### 3. ОНОЛЫН СУДАЛГАА

#### 3.1 Санах ойн хаяг

Програм ажиллах явцад объект бүр (хувьсагч болон хүснэгт) санах ойн орчимд хаа нэгтэй байрладаг. Санах ой дахь объектын байрлалыг түүний хаяг (address) гэдэг. Санах ойд хаяг олгосноор эдгээр хаягаар дамжуулж програм хангамжийн бүхий л түвшинтэй харилцах боломж олгодог. C++ хэлэнд объектын хаягыг олж авхад зориулсан хаяг оператор & байдаг.

#### 3.2 Заагч санах ойд байрлах хэмжээ

Заагч нь санах ойд байрлахдаа 32 битийн үйлдэлийн системд 32 бит, 64 битийн үйлдэлийн системд 64н битийг эзлэн байрладаг. Заагчын хэмжээ нь тухайн заагчын төрлөөс үл хамааран адилхан хэмжээтэй байдаг. Учир нь заагч дотор өөр хувьсагчийн хаягыг хадгалж байдаг болхоор хангалттай том байх ёстой. Заагч бол санах ойн хаягыг зааж байдаг хувьсагч юм.

#### 3.3 Хаяган хувьсагч

Хаягаар дамжуулан бид санах ойд байрлах хүссэн хувьсагчидаа хандах боломжтой байдаг. Хаяган хувьсагчийг зарлахын тулд \* оператор ашигладаг(int \*a;). Харин хувьсагчийн

санах ойн хаягыг хаяган хувьсагчид авахдаа & оператор ашигладаг. (\*a=&b;)

Хаягыг бүх төрлөөр зарлах боломжтой байдаг.

### 3.4 Заалтан хувьсагч

Тухайн хувьсагчийн санах ойн утгыг өөр нэрээр дуудан ажилуулах боломжтэй хувьсагч. Энэ нь хаяган хувьсагчийг бодвол амар ба санах ойд шууд хандаж чаддаг.

Int a=4;

Int& k=4 гэж ашиглах ба a=k адил утгатай болно.

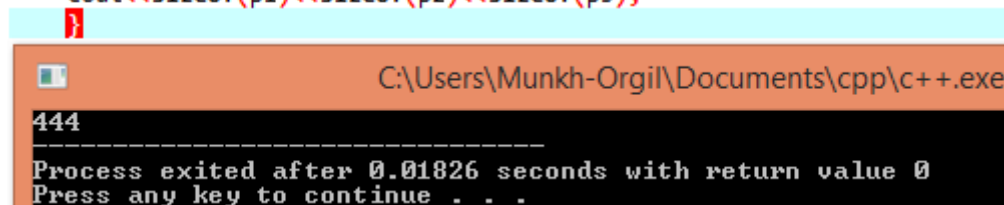
### 3.5 Санах ойн цоорхой

Санах ой дээр ахин ашиглах боломжгүй нөөцлөөд орхисон санах ой үүсгэхийг санах ойн цоорхой гэнэ.

## 4. ХЭРЭГЖҮҮЛЭЛТ

1. Хаяган хувьсагчийн кодыг хэрэгжүүлсэн компьютер 32 бит үйлдэлийн системтэй байсан учир хаяган хувьсагч санах ойд 32 бит буюу 4 байт болж 4 4 4 г хэвэлсэн.

```
#include <iostream>
using namespace std;
int main()
{
    char *p1;
    int *p2;
    double *p3;
    cout<<sizeof(p1)<<sizeof(p2)<<sizeof(p3);
}
```



2. **int a=125;**            а хувьсагчид 125 гэсэн тоог хадгалж байна

**int \*p = &a;**            а гийн хаягыг р хаяган хувьсагчид хийж байна.

**cout<<p;**                а- г хадгалагдсан хаяган хувьсагчийг 16таар хэвлэнэ.

**cout<<\*p;**              р -д хадаглагдсан а гийн утгыг хэвлэнэ.

**p++;**                    Хаягыг байршилыг 1р нэмж байна.

**cout<<p;**                Түүний хаягыг хэвлэж харуулна.

**cout<<\*p;**              Түүнд оноогдох утгыг хэвлэж харуулна. Үүнд гараас утга өгөөгүй учир compiler санамсаргүй утга өгнө.

3. **int numbers[5];** 5н тоон элемент агуулах хүснэгт зарлаж байна.

**int \*p;** p хаяган хувьсагч зарлаж байна.

**p = numbers; \*p = 10;** numbers хүснэгтийн эхний элементийн хаягыг p хаяган хувьсагчид байрлалд өгж байна. Мөн p хаяган хувьсагчийн утганд 10г оноосон.

**p++; \*p = 20;** p хаяган хувьсагчийн байрлалыг 1p нэмэн түүний утганд 20г өгсөн.

**p = &numbers[2]; \*p = 30;** numbers хүснэгтийн 3р элементийн байрлалыг p хувьсагчид хадгалж түүний утгыг 30 болгон өөрчилсөн.

**p = numbers + 3; \*p = 40;** numbers хүснэгтийн эхний элементийн хаягыг 3аар нэмэгдүүлж, эхний элементээс хойших 3дах элементийн хаягын утгыг 40 болгон өөрчилсөн.

**p = numbers; \*(p+4) = 50;** numbers хүснэгтийн эхний элементийн хаягыг p хаяган хувьсагчид хадгалж, p ийн эхний байрлалаас хойших 4дөх элемент буюу хүснэгтийн 5р байрлалын утгыг 50 болгож байна.

**for (int n=0; n<5; n++)** Давталт ашиглан хүснэгтийн утгуудыг хэвлэж байна.  
**cout << numbers[n] << ", ";**

4. Функцийн аргументээр хаяган хувьсагч дамжуулж өгсөнөөр програм хангамжийн бүх түвшинд бид ажиглаж болдогын жишээ. Хаяган хувьсагч дамжуулж өгөхдөө аргументэнд хувьсагчыг \*, & операторууд ашиглан дамжуулж болдог. Хоёр тооны утгыг солих функц бичэхдээ гараас авсан хувьсагчын утгын хаягыг & операторыг ашиглан аргументэнд дамжуулж өгсөн. Тухайн хэрэглэгчийн функцээр дамжиж ирсэн хаягыг хоорондоо солихдоо \* оператор ашиглан ирсэн хаягын утгыг авч хооронд сольсон. Солих үйлдэлийн тухайд утга хадгалах зориулттай нэг хувьсагч авж түүндээ функцийн аргументээр ирсэн эхний хаяган хувьсагчийн утгыг тэнцүүлж, солих хувьсагчийг эхний хувьсагчтай тэнцүүлж хадгалсан хувьсагчийг солигдах хувьсагчид хийсэн.

```
void change(int *a, int *b){
    int s=*a;
    *a=*b;
    *b=s;
}
```

5. Заалтан хувьсагч ашиган бодох бодлого. Заалтан хувьсагч гэдэг бол хувьсагчийн нөгөө нэр учир функцээр дамжуулахдаа заалтан хувьсагч ашигласан. Давуу тал нь өөрийн хэрэгтэй хувьсагчийг өөрт хэрэгтэй нэрээр дуудан ажилуулах боломжтэй. Үүнийг ашиглан 2 хувьсагчийн утгыг солих функц бичсэн.

Хувьсагчид

```
int a, b;
```

Заалтан хувьсагч

```
void change(int &x, int &y)
```

## 5. ДҮГНЭЛТ

Санах ойг ашигласнаар C++ хэлэнд олон давуу талууд гарж ирдэг. Заагчин хувьсагчийг ашигласнаар хүснэгтээр хийхэд төвөгтэй байдаг элемент хасаж, нэмэх, шилжүүлэх үйлдэлүүдийг гүйцэтгэж болдог. Мөн хаяган хувьсагчийг ашигласнаар энгийн арифметик үйлдэл хийж болно.

Заалтан хувьсагч бидэнд заагчаас арай хөнгөн, санах ойд шууд ханддаг, ойлгоход хялбар, санах ойн утгуудыг ашиглаж чаддагаараа бидэнд давуу тал олгож байна.

## 6. АШИГЛАСАН МАТЕРИАЛ

1. <https://mathbits.com/MathBits/CompSci/Pointers/Addresses.htm>
2. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.
3. <https://en.cppreference.com/w/c>

## 7. Хавсралт

### Хаяган хувьсагч

```
#include<iostream>
```

```
using namespace std;
```

```
void change(int *a, int *b);
```

```
int main(){
```

```
    int i,j;
```

```
    cout<< "ehnii i: ";
```

```
    cin>>i;
```

```
    cout<< "ehnii j: ";
```

```
    cin>>j;
```

```
    change(&i, &j);
```

```
        cout<< "daraagin i: "<<i;
```

```
        cout<< "\ndaragin j: "<<j;
```

```

}

void change(int *a, int *b){
    int s=*a;
    *a=*b;
    *b=s;
}

void change(int *a, int *b){
    int s=*a;
    *a=*b;
    *b=s;
}

```

### **Заалтан хувьсагч**

```

#include<iostream>
using namespace std;
void change(int &x,int &y){
    int s;
    s=x;
    x=y;
    y=s;
}

int main(){
    int a, b;
    cin>>a;
    cin>>b;
    cout<<"ehnii a= "<<a<<endl;
    cout<<"ehnii b= "<<b<<endl;
}

```

```
change(a,b);  
cout<<"a= "<<a<<endl;  
cout<<"b= "<<b<<endl;  
}
```