# (Лаборатори №5)

ХШУИС, МКУТ

Г.Мөнх-Оргил 18b1num2048

# 1. ОРШИЛ

Обьект байгуулахын тулд C++ -н компайлер тодорхой ажил хийдэг, тухайлбал гишүүн өгөгдөлд ой бэлдэж түүнд гарааны утга оноодог бөгөөд үүнтэй адил байгуулагч функцуудыг энэ ажлын хүрээнд авч үзэх болно.

# 2. ЗОРИЛГО

1. Байгуулагч функц хэзээ дуудагддаг вэ?

2. Устгагч функц хэзээ дуудагдах вэ?

3. Хуулагч байгуулагч гэж юу вэ?  Ач холбогдол нь юу вэ?

4. Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?

5. Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? new оператороор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

# 3. ОНОЛЫН СУДАЛГАА

## **3.1 Байгуулагч функц хэзээ дуудагддаг вэ?**

Обьект байгуулагдах үед түүнд зориулан ой бэлдэхээр анхдагч эсвэл хэрэглэгчийн тодорхойлсон байгуулагч функц үргэлж дуудагдана. Тйимээс ямар нэгэн байгуулагч функц дуудахгүйгээр обьект байгуулагдахгүй.

## **3.2 Устгагч функц хэзээ дуудагдах вэ?**

Обьектэд зориулан байгуулагч функцийн бэлдэх ойг хэрэглэж дууссаны дараагаар чөлөөлөхөд устгагч функцийг хэрэглэнэ. Устгагч функц нь цаашдаа хэрэглэгдэхгүй обьект устахад түүний эзэмшиж байсан ойг чөлөөлнө. Хэрэв класс дотор устгагч функцийг илээр тодорхойлж өгөөгүй тохиолдолд обьект устахад анхдагч устгагч функцийг C++ компайлер нэмж тодорхойлж хэрэглэнэ.

## **3.3 Хуулагч байгуулагч гэж юу вэ?**

Шинэ обьект өмнөх аль нэгэн обьектын тухайн үеийн утгыг гарааны утга болгон авахыг хуулагч байгуулагчаар шийдэж болно. Хуулагч байгуулагч нь обект хуулбарлах функц юм.

**3.4** Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?

Хуулагч байгуулагч бол шинэ обьект нь түүний өмнөх аль нэгэн обьектын сүүлийн утгыг хуулбарлаж авах боломжийг олгох функц юм. Ийм маягаар байгуулагдах шинэ обьект ба утга нь хуулбарлагдах обьект хоорондоо зөвхөн нэрээрээ ялгаатай.

Ийм байгуулагч нэг параметр авах ба тэр нь класс нэртэй хуулагдах обьектын заалт байна.

**3.5** Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? new оператороор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

обьектийн хаяган хувьсагчийг дээрх байдлаар зарлаж болно.

employee \*pptr;

employee emp;

pptr = &emp

обектон хаяган хувьсагчаар дамжуулж гишүүн функцэд хандах

* Гишүүнчлэлийн цэг (**.**) оператор хэрэглэх

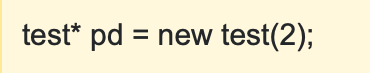
(\*pptr)**.**getdata ();

(\*pptr)**.**showdata ();

* Гишүүнчлэлийн сум (→) оператор хэрэглэх

pptr→ getdata ();

pptr→ showdata();



New түлхүүр үг шинээр үүсэн обьектыг санах ой дээр байрлуулдаг. Энэ нь С хэлний Malloc тай адилхан ажиллаж, дараа нь бидэнд утсгах боломж олгодог.

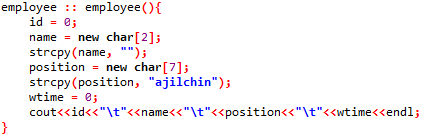


New түлхүүр үг ашиглахгүй шууд зарлах үед обьект нь санах ой дээр шууд байрладаггүй ба санах ойтэй холбох ажилгааг компайлер хийдэг. Классын хүрээнээс гарахад устдаг эсвэл мартагддаг.

4. Хэрэгжүүлэлт

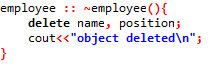
1. new оператор ашиглан санах ой нөөцлөх

employee төрлийн класс-д классын нэртэй адил функц үүсгэсэн. Обьект үүсэхэд анхны утгууд нь гишүүн өгөгдөл тус бүрийг утгатай байлгахаар зарлаж өгсөнийг доорх жишээнээс харж болно.



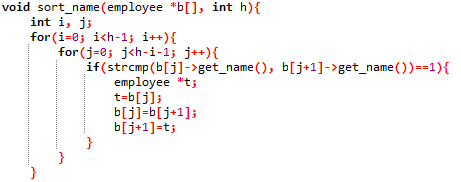
2. Санах ой чөлөөлөх

Гишүүн өгөгдөлд хоёр char төрлийн хувьсагч байгаа бөгөөд обьект үүсэн ашиглагдаад дуусах үед санах ой дээр ойн цоорхой үүсэх учир классын нэртэй адил нэртэй урдаа ~ оператор залган байгуулан, тухайн char төрлийн гишүүн өгөгдөлүүдийг хаягаар нь хандан устгах функц бичсэн. Програм ажиллаж дуусах үед устгагч функц ажиллаж гишүүн өгөгдөлүүдийг устгасанаа хэвлэсэн.



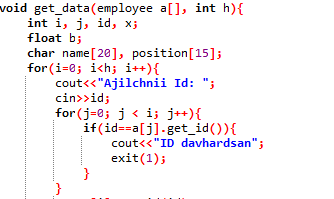
3. Нэрээр эрэмбэлэх

Хүснэгтээр обьектуудыг дамжуулж, дамжуулагдсан обьектуудын нэрийг эхнээс нь сонгон авж strcmp функц ашиглан хооронд жиших замаар нэрээр эрэмбэлсэн.



4. ID шалгах

Хүснэгтээр ажилчдын id-ийг дамжуулж, өмнөх id-тай тэнцүү эсэхийг шалгах замаар гүйцэтгэсэн.



# 5. ДҮГНЭЛТ

# Энэхүү лабораторийн ажлаар байгуулагч функц, устгагч функц гэж юу болох, эдгээрийг хэрхэн хэрэглэх талаар судалж мэдлээ. Байгуулагч функц тодорхойлсноор классын обьектод санаах ой нөөцлөх, гарааны утга оноох ажлуудыг хийж болох ба ямар нэгэн байгуулагч функц дуудагдахгүйгээр обьект байгуулагдахгүй, үүнтэй адилаар обьект нь устгагч функцгүйгээр устаж чадахгүй.

1. Ашигласан материал
2. <https://www.geeksforgeeks.org/constructors-c/>
3. https://www.tutorialspoint.com/cplusplus/cpp\_constructor\_destructor.html
4. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар
5. Хавсралт

#include<iostream>

#include<string.h>

#include<conio.h>

#include<windows.h>

#include<bits/stdc++.h>

using namespace std;

class employee{

private:

int id;

char \*name;

char \*position;

float worked\_time;

int zahirliin\_tsalin();

public:

employee();

employee(int a, char \*b, char \*c, float d);

void set\_id(int a);

int get\_id();

void set\_name(char \*a);

char\* get\_name();

void set\_position(char \*a);

char\* get\_position();

void set\_worked\_time(float a);

float get\_worked\_time();

int tsalin\_bodoh();

~employee();

};

employee :: employee(){

id = 0;

name = new char[2];

strcpy(name, "");

position = new char[7];

strcpy(position, "ajilchin");

worked\_time = 0;

cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<worked\_time<<endl;

}

employee :: employee(int a, char \*b, char \*c, float d){

id=0;

name = new char (strlen(name));

strcpy(name, b);

position = new char(strlen(position));

strcpy(position, c);

worked\_time = 0;

cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<worked\_time<<endl;

}

employee :: ~employee(){

delete name, position;

cout<<"object deleted\n";

}

void employee :: set\_id(int a){

id=a;

}

int employee :: get\_id(){

return id;

}

void employee :: set\_name(char \*a){

name = new char(strlen(a)+1);

strcpy(name, a);

}

char\* employee :: get\_name(){

return name;

}

void employee :: set\_position(char \*a){

position = new char(strlen(a)+1);

strcpy(position, a);

}

char\* employee :: get\_position(){

return position;

}

void employee :: set\_worked\_time(float a){

worked\_time=a;

}

float employee :: get\_worked\_time(){

return worked\_time;

}

int employee :: zahirliin\_tsalin(){

return worked\_time\*22\*5000;

}

int employee :: tsalin\_bodoh(){

int a, b;

if(strcmp(position, "zahiral")!=0){

a=worked\_time\*22;

b=a\*3750;

return b;

}

else

return zahirliin\_tsalin();

}

void get\_data(employee a[], int h){

int i, j, id, x;

float b;

char name[20], position[15];

for(i=0; i<h; i++){

cout<<"Ajilchnii Id: ";

cin>>id;

for(j=0; j < i; j++){

if(id==a[j].get\_id()){

cout<<"ID davhardsan";

exit(1);

}

}

a[i].set\_id(id);

cout<<"Ajilchnii ner: ";

cin>>name;

a[i].set\_name(name);

cout<<"Ajilchnii alban tushaal: ";

cin>>position;

a[i].set\_position(position);

cout<<"Ajilchnii ajillasan tsag: ";

cin>>b;

a[i].set\_worked\_time(b);

}

}

void print(employee a[], int h){

int i;

for(i=0; i<h; i++){

cout<<a[i].get\_id()<<"\t";

cout<<a[i].get\_name()<<"\t";

cout<<a[i].get\_position()<<"\t\t";

cout<<a[i].get\_worked\_time()<<"\t";

cout<<a[i].tsalin\_bodoh()<<"\n";

}

}

void sort\_name(employee \*b[], int h){

int i, j;

for(i=0; i<h-1; i++){

for(j=0; j<h-i-1; j++){

if(strcmp(b[j]->get\_name(), b[j+1]->get\_name())==1){

employee \*t;

t=b[j];

b[j]=b[j+1];

b[j+1]=t;

}

}

}

cout<<"sorted name: "<<endl;

for(i=0; i<h; i++){

cout<<b[i]->get\_id()<<"\t";

cout<<b[i]->get\_name()<<"\t";

cout<<b[i]->get\_position()<<"\t\t";

cout<<b[i]->get\_worked\_time()<<"\t";

cout<<b[i]->tsalin\_bodoh()<<"\n";

}

}

int main(){

int h;

cout<<"Heden ajilchin burtgeh ve: "<<endl;

cin>>h;

employee a[h];

get\_data(a, h);

employee \*b[h];

int i;

for(i=0; i<h; i++)

b[i]=&a[i];

sort\_name(b, h);

}