# (Лаборатори №4)

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
2. Анхдагч байгуулагч функц
3. Устгагч функц
4. Функц дахин тодорхойлох
5. Данамик санах ой (new, delete oператор ашигласан) болон байгуулагч, устгагч функцийг хамтад нь ашиглах
6. Хандалтын түвшин

# 3. ОНОЛЫН СУДАЛГАА

## **3.1 Байгуулагч функц (Constructor function)**

Классын обьект байгуулах үед санах ой нөөцлөх, гарааны утга оноох зорилготой тусгай зориулалттай функцийг байгуулагч функц гэнэ. Обьектод ой бэлдэж түүний гишүүн өгөгдөлд гарааны утга оноож өгөхийн тулд систем байгуулагчийг дууддаг. Уг функц ямар ч утга буцаадааггүй, класстайгаа ижил нэртэй байна. (3)

## **3.2 Анхдагч байгуулагч функц (Default Constructor function)**

Хэрэв програм хөгжүүлэгч ямар нэгэн байгуулагч тодорхойлж өгөөгүй байвал C++ компайлер өөрөө ийм функцийг програмд нэмж тодорхойлдог. Ийм байгуулагчийг default constructor буюу анхдагч байгуулагч гэнэ. (3)

Анхдагч байгуулагч функц бол классын обьектод анхны утга оноох, санах ой нөөцлөдөг онцгой гишүүн функц юм. Анхдагч байгуулагч функц нь обьект үүсэх үед автоматаар үүсгэгддэг. Классынхаа нэртэй адилханаар зарлагддаг. Ямар нэгэн параметр авдаггүй.

https://www.geeksforgeeks.org/constructors-c/

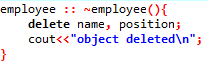
## **3.3 Устгагч функц ( Destructor function)**

Устгагч функц нь байгуулагч функцээр үүссэн обьектыг устган санах ойг чөлөөлөхөд ашиглагддаг. Устгагч функц нь байгуулагч функцтэй адилхан классынхаа нэрийг авдаг ба функцын нэрний өмнө (~) аван ялгаж өгдөг. Устгахдаа гишүүн өгөгдөлийн хаягаар нь хандаж устгадаг.

Ямар нэгэн параметр авахгүй мөн утга буцаахгүй. Програм зогсоход дуудагдаж устагдаг.

Хэрэгжүүлэхдээ дараах байдлаар хэрэгжүүлнэ.

https://www.tutorialspoint.com/cplusplus/cpp\_constructor\_destructor.html



**3.4 Функц дахин тодорхойлох**

Функцийн нэр нь ижил, нэгэн төрлийн боловч авах аргументийн тоо, төрөл, дараалал нь ялгаатай олон янзын функцийг тодорхойлох үйлийг функц дахин тодорхойлох гэх ба харин ийм функцийг дахин тодорхойлогдох функц гэнэ. (3)

Жишээ нь:

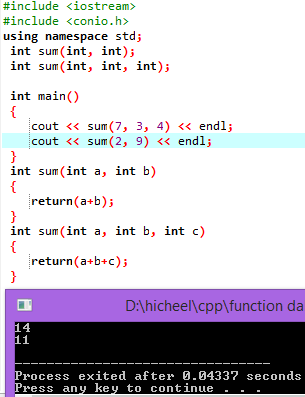
2, 3 тооны нийлбэр олох 2 өөр функцийг Си хэлээр доор үзүүлсэн шиг бичих шаардлагатай болно.

int sum2(int a, int b);

int sum3(int a, int b, int c );

Энэ sum2(), sum3() хоёр функц хоёулаа өгөгдсөн тоонуудын нийлбэр олох үүрэгтэй боловч өөр өөр нэртэй байна.

Харин C++ хэлэнд



**3.5 Динамик санах ой (new, delete оператор ашигласан) болон байгуулагч, устгагч функцийг хэрхэн хамтад нь ашиглах вэ?**

Байгуулагч функцээр тодорхойлогдсон гишүүн өгөгдөлүүд обьект үүсгэн ашиглагдаад дуусах үед санах ой дээр ойн цоорхой үүсгэх учир үүнээс сэргийлж, устгагч функц зарлан, функц дотроо delete оператор ашиглан ашиглагдаад дууссан гишүүн өгөгдлүүдийг санах ойгоос чөлөөлөх боломжтой байдаг.

**3.6 Хандалтын түвшин**

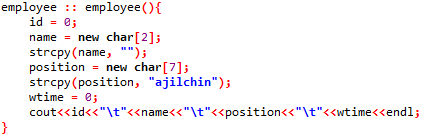
Обьектын гишүүн өгөгдөлд обьектээр нь дамжуулж хандах эсэхийг шийднэ.

* Private- классын гишүүн өгөгдөлд обьектээр нь дамжуулж хандахыг хориглоно
* Public- обьектээр нь дамжуулж гишүүн өгөгдөлд хандахыг зөвшөөрнө
* Protect- гишүүдэд классын гаднаас хандаж болохгүй, гэхдээ удамшсан классуудаас хандах боломжтой. (3)

# 4. ХЭРЭГЖҮҮЛЭЛТ

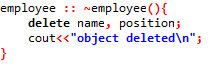
1. Байгуулагч функц

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.



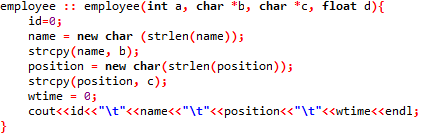
2. Устгагч функц

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.



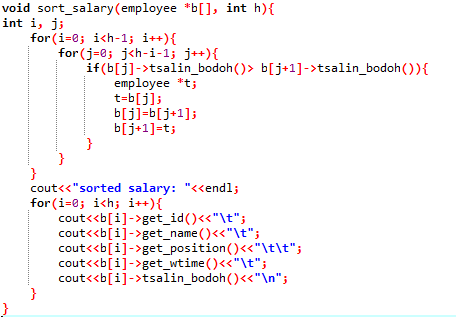
3. Параметртэй байгуулагч

Ялгаатай обьект байгуулах зорилгоор параметртэй байгуулагч бичсэн ба классын нэртэй адил нэртэй параметр авж чадах байгуулагч функц юм.



4. Цалингаар эрэмбэлэх

Хүснэгтээр обьектуудыг дамжуулж, цалин бодох функцийг ашиглаж бодогдсон цалинг хооронд нь жиших замаар цалингаар эрэмбэлсэн.



# 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 wtime;

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\_wtime(float a);

float get\_wtime();

int tsalin\_bodoh();

~employee();

};

employee :: employee(){

id = 0;

name = new char[2];

strcpy(name, "");

position = new char[7];

strcpy(position, "ajilchin");

wtime = 0;

cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<wtime<<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);

wtime = 0;

cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<wtime<<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\_wtime(float a){

wtime=a;

}

float employee :: get\_wtime(){

return wtime;

}

int employee :: zahirliin\_tsalin(){

return wtime\*22\*5000;

}

int employee :: tsalin\_bodoh(){

int a, b;

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

a=wtime\*22;

b=a\*3750;

return b;

}

else

return zahirliin\_tsalin();

}

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

int i, j;

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

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

if(b[j]->tsalin\_bodoh()> b[j+1]->tsalin\_bodoh()){

employee \*t;

t=b[j];

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

b[j+1]=t;

}

}

}

cout<<"sorted salary: "<<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";

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

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

}

}

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

int i, id;

float b;

char name[20], position[15];

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

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\_wtime(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\_wtime()<<"\t";

cout<<a[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\_salary(b, h);

}