

Ажилчин Классыг Дахин Загварчилж, Байгуулагч Болон Устгагч Функц Тодорхойлох (Лаборатори №4)

Н. Энхболд

ХШУИС, Програм хангамжийн 2-р түвшний оюутан, 20B1NUM0102

1. ОРШИЛ/УДИРТГАЛ

Өмнөх лабораторийн ажиллыг шинэчилж Constructor болон destructor функцийг үр дүнтэйгээр хэрэглэн олон ажилчдын цалинг бодов. Үүний тулд классын хүснэгт үүсгэж олон объект болох ажилчдыг үүсгэв.

2. ЗОРИЛГО

“Ажилчин” гэх классыг энэ удаад Байгуулагч болон Устгагч функцийн тусламжтайгаар дахин загварчилж түүндээ new/delete операторуудыг хэрхэн ашиглахыг жишээ болгон харуулна. Мөн ажилчин классын pointer буюу хаяган хувьсагч үүсгэж түүндээ хэрхэн динамик санах ой үүсгэн нэг дор олон объект дээр тооцоолол хийж харуулна.

3. ОНОЛЫН СУДАЛГАА

3.1 Байгуулагч функц

Аливаа C++ -ийн класст “constructor” буюу байгуулагч гэх функц заавал байх ёстой бөгөөд энэхүү функц нь объектыг байгуулах/үүсгэх үүрэгтэй. Constructor функц нь ямар нэгэн утга буцаадгүй бөгөөд классынхаа нэртэй ижил нэрээр тодорхойлдог. Мөн энэхүү функцийг авах параметрээр нь 4 ангилдаг:

- Анхдагч байгуулагч

Ямар нэгэн параметр авдаггүй бөгөөд хэрэв класс дотор ямар нэг constructor бий болгон тодорхойлоогүй тохиолдолд C++ compiler ажиллах үедээ анхдагч байгуулагчийг нэмж өгдөг.

- Параметртэй байгуулагч

Энэхүү функц нь тодорхой төрлийн 1 бөгөөд түүнээс олон параметр авдаг байж болох бөгөөд тэрхүү утгуудаа өөр дээрээ боловсруулах үйлдэл хийдэг. Жишээ нь: Тухайн утгуудаа ашиглан гишүүн өгөгдөлдөө утга оноох зэрэг.

- Хуулагч байгуулагч

Мөн адил параметр авдаг боловч утган дээрээ объект авдгаараа онцлог. Бусад байгуулагчийн адилаар класстайгаа ижил нэртэй, утга буцаадаггүй байна. Параметр дээрээ хуулбарлах объектынхоо хаягийг авдаг боловч байгуулагдсаныхаа дараачаар

тухайн объектын гишүүдийн утгыг өөрчлөхөд тухайн хаягаа дамжуулсан объектын утга өөрчлөгддөггүй.

- Анхны утга бүхий параметртэй байгуулагч

Уг байгуулагч функц нь Параметрт байгуулагчтай адилаар параметр авдаг боловч хэрэв параметр дамжуулж өгөөгүй тохиолдолд өөрт буй гарааны утгаа авдаг. Мөн аливаа анхдагч байгуулагчийг бүх параметр нь гарааны утгатай Анхны утга бүхий бүхий параметртэй байгуулагчтай хамт хэрэглэж болохгүй.

3.2 Устгагч функц

Объектыг хэрэглэж дуусаад устгаж ойг чөлөөлөх үүрэгтэй. Устгагч буюу Destructor функц нь класст мөн адилаар заавал байх ёстой бөгөөд хэрэв тодорхойлж өгөөгүй тохилдолд C++ compiler ажиллах үедээ нэмж өгдөг. Constructor функцээс ялгаатай нь класст зөвхөн ганц л Устгагч функц байх ёстой байдаг ба тодорхойлохдоо нэрнийхээ өмнөө “~” тэмдэгт авдаг.

3.3 Функц дахин тодорхойлох

Функцийн нэр нь адилхан боловч авах параметрээрээ ялгарч буй функцууд 1 дор оршин байх боломжтой. Үүнийг Полиморфизм гэх ба эдгээр ижил нэртэй функцуудаас тодорхой 1-ийг нь дуудахдаа параметрийг нь тааруулан өгөх ёстой (Polymorphism, n.d.). Constructor функц дээр жишээ авбал:

```
class Employee{
private:
    int id;
    char *name;
public:
    Employee() {
        *name = new char[20];
    }
    Employee(int a, char b[]){
        Id = a;
        *name = new char[20];
        strcpy(name, b);
    }
}
```

```
}
```

Хэрэв Employee a(2, “Bold”) дуудах тохиолдолд параметр бүхий 2 дох функцийг дуудах ба ямар нэгэн утга өгөлгүй Employee a; гэж объект үүсгэвэл эхний функцыг ажиллуулна.

3.4 Динамик санах ой болон байгуулагч функц, устгагч функцийг хэрхэн хамтад нь ашиглах вэ?

Динамик санах ой буюу new/delete операторууд нь санах ой бэлдэх, санах ой чөлөөлөх үүрэгтэй ба эдгээр операторуудыг байгуулагч болон устгагч функцууд дээр санах ойн цоорхой үүсгэлгүйгээр ашиглаж болдог. Жишээ нь:

Класс дотор char *name гэх хаяган хувьсагч байвал түүнд санах ой бэлдэж өгөх хэрэгтэй. Үүнийг constructor функцдээ хийж өгөх ба

```
Employee(){name = new char[20]}
```

зэргээр ашиглах ба энэхүү бэлдсэн санах ойгоо хэрэглэж дуусаад тухайн объектод нь хамт устгаж болно. Destructor функц тодорхойлохдоо түүндээ delete операторыг

```
~Employee(){delete name}
```

Зэргээр дотор нь ашиглан санах ойн цоорхойгоос сэргийлэх боломжтой юм.

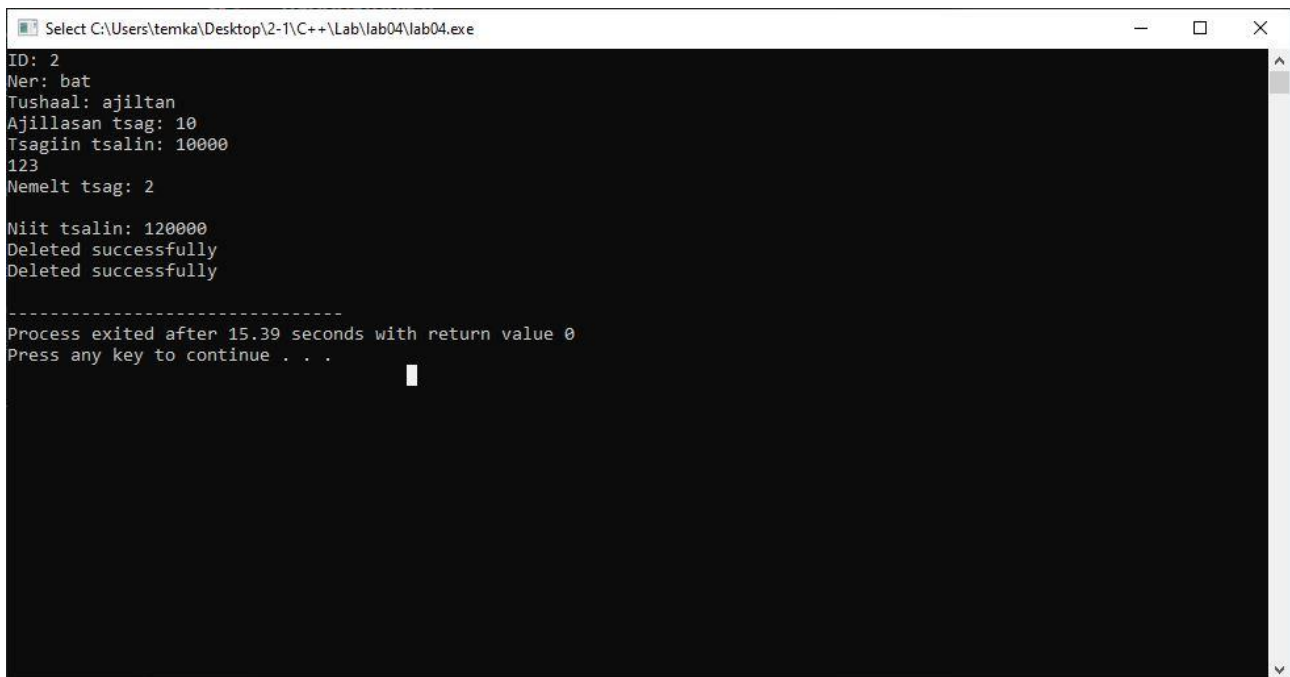
4. ХЭРЭГЖҮҮЛЭЛТ

1. Класст анхдагч болон параметртэй багуулагч нэмж тодорхойлсон нь:

```
17 Employee(){
18     name = new char[20];
19     position = new char[10];
20 }
21 Employee(int tsagiinHuls){
22     name = new char[20];
23     position = new char[10];
24     workHour = tsagiinHuls;
25 }
26 Employee(int tsagiinHuls, int dugaar, char *ner, char *tushaal, float tsag)
27 {
28     name = new char[20];
29     position = new char[10];
30     id = dugaar;
31     strcpy(name, ner);
32     strcpy(position, tushaal);
33     workHour = tsag;
34 }
35 ~Employee(){
36     delete name;
37     delete position;
38 }
```

Тодорхойлохдоо name болон position буюу нэр болон албан тушаал гэх гишүүдэд санах ой бэлдэж өгж байгаа бөгөөд энэхүү санах ойгоо Destructor функцдээ устгаж өгж байгаа билээ.

2. Устгагч функц нь хэзээ устаж байгаа тодорхойлох:



```
Select C:\Users\temka\Desktop\2-1\C++\Lab\lab04\lab04.exe
ID: 2
Ner: bat
Tushaal: ajiltan
Ajillasan tsag: 10
Tsagiin tsalin: 10000
123
Nemelt tsag: 2

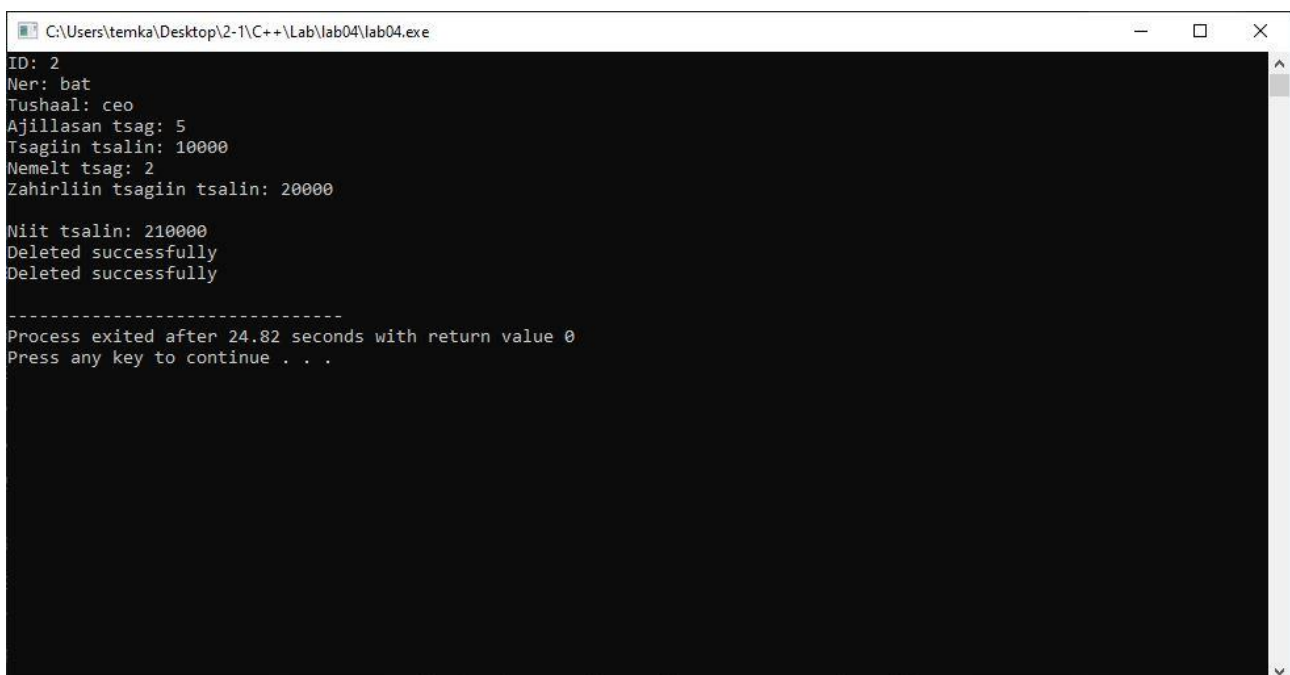
Niit tsalin: 120000
Deleted successfully
Deleted successfully

-----
Process exited after 15.39 seconds with return value 0
Press any key to continue . . .
```

Устгагч функц дээрээ `cout << "Deleted successfully"`; командыг нэмж өгсөн бөгөөд програм ажиллаад дуусах үед байгуулсан `"bat, bold"` 2 объектыг амжилттай устгалаа гэж дэлгэцэнд хэвлэж байна.

3. Захирлын цалин бодох функцийг `private` хандалтын түвшинтэй болгож цалин бодох функц дотроо дуудан нийт цалинг олох:

Хэрвээ албан тушаал нь `"ceo"` буюу захирал байвал:

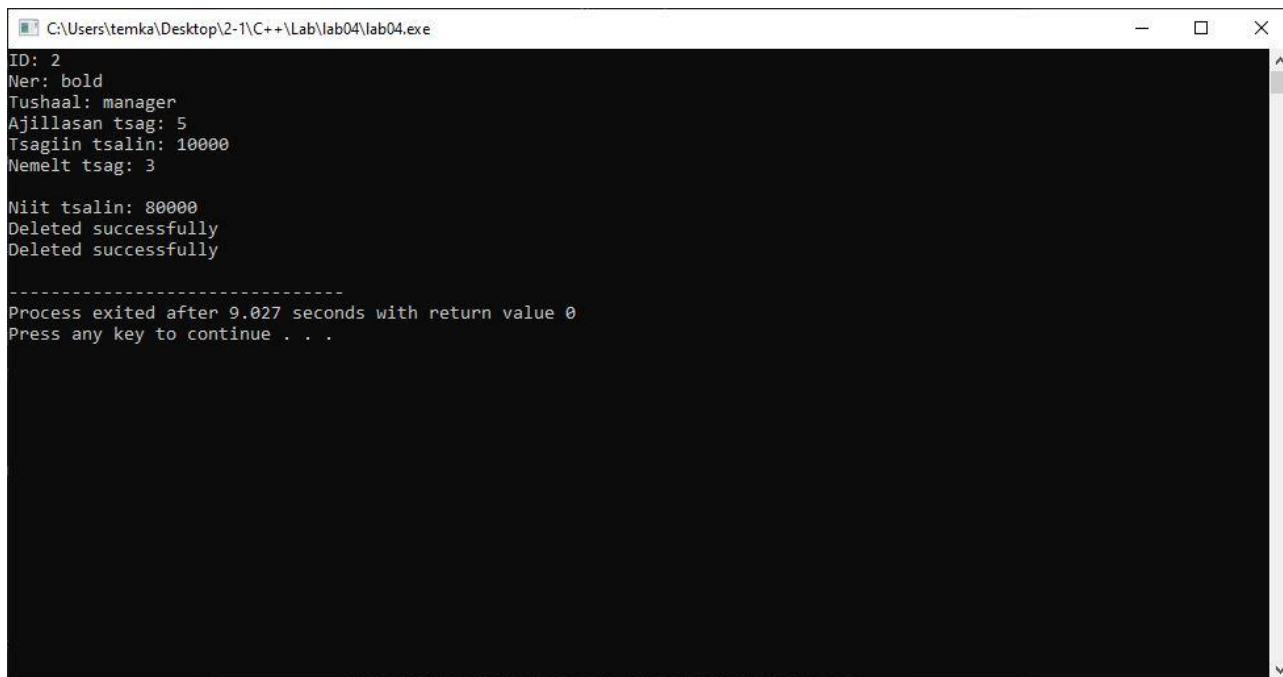


```
C:\Users\temka\Desktop\2-1\C++\Lab\lab04\lab04.exe
ID: 2
Ner: bat
Tushaal: ceo
Ajillasan tsag: 5
Tsagiin tsalin: 10000
Nemelt tsag: 2
Zahirliin tsagiin tsalin: 20000

Niit tsalin: 210000
Deleted successfully
Deleted successfully

-----
Process exited after 24.82 seconds with return value 0
Press any key to continue . . .
```

Хэрвээ бусад ажилтан байвал:



```
C:\Users\temka\Desktop\2-1\C++\Lab\lab04\lab04.exe
ID: 2
Ner: bold
Tushaal: manager
Ajillasan tsag: 5
Tsagiin tsalin: 10000
Nemelt tsag: 3

Niit tsalin: 80000
Deleted successfully
Deleted successfully

-----
Process exited after 9.027 seconds with return value 0
Press any key to continue . . .
```

Гэх зэргээр цалин бодох функц дотроо шалган цалинг харуулав.

4. Бүх гишүүн өгөгдөл private үед setter, getter функц ашиглан утга оноож, буцаан утгыг авах:

```
Void Employee::setter(int dugaar, char *ner){
```

```
    Id = dugaar;
```

```
    Strcpy(name, ner);
```

```
}
```

```
Int Employee::getter(){
```

```
    Return id;
```

```
}
```

Гэх зэргээр private гишүүдэд хандаж, утгыг авч болно.

5. Классын хүснэгт ашиглан олон ажилчны утга оруулж Нийт цалингуудыг хооронд нь эрэмбэлэх:

```
int main(){
    int total[5];
    Employee *ajilchid = new Employee[5];
    for(int i = 0; i < 5; i++){
        (ajilchid+i)->input();
        *(total+i) = (ajilchid+i)->salary();
        cout << endl << "Niit tsalin: " << *(total+i) << endl;
    }
    // ajilchid->input();
    // total = ajilchid->salary();
    // cout << endl << "Niit tsalin: " << total << endl;
    // (ajilchid+1)->input();
    // total = (ajilchid+1)->salary();
    // cout << endl << "Niit tsalin: " << total << endl;
    int temp, j, i;
    for(i = 0; i < 5; i++){
        for(j = i + 1; j < 5; j++){
            if(total[j] < total[i]){
                temp = total[i];
                total[i] = total[j];
                total[j] = temp;
            }
        }
    }
    for(i = 0; i < 5; i++){
        cout << total[i]<< endl;
    }
    delete ajilchid;
}
```

```
Select C:\Users\temka\Desktop\2-1\C++\Lab\lab04\lab04.exe
ID: 1
Ner: bat
Tushaal: manager
Ajillasan tsag: 2
Tsagiin tsalin: 10000
Nemelt tsag: 3

Niit tsalin: 50000
ID: 2
Ner: bold
Tushaal: jijvr
Ajillasan tsag: 4
Tsagiin tsalin: 10000
Nemelt tsag: 3

Niit tsalin: 70000
ID: 3
Ner: bazraa
Tushaal: ceo
Ajillasan tsag: 4
Tsagiin tsalin: 15000
Nemelt tsag: 2
Zahirliin tsagiin tsalin: 10000

Niit tsalin: 150000
50000
70000
150000
Deleted successfully

-----
Process exited after 49.14 seconds with return value 3221226356
Press any key to continue . . .
```

5. ДҮГНЭЛТ

C++ compiler-т constructor/destructor функцээ даалган тодорхойлохгүйгээс үүдэн маш олон үйлдлийг илүүчилж 3-р лаборатори дээр хийсэн билээ. Динамик санах ойн операторуудыг эдгээр функцууд дээрээ ашигласнаар санах ойн цоорхой, массивын багтаамж хүрэхгүй байх зэрэг олон дутагдлуудаас зайлс хийж чадаж байгаа билээ. Мөн олон өгөдөлд хандаж болохоос илүүтэй хэрэглэгчид амар байдлаар public хандах горимтой гишүүн функцдаа бусад private функцуудаа дуудаж өгсөн. Гишүүд бие биетэйгээ уялдаатай тохиолдолд үүнд зориулан ганц хандах боломжтой функцтай байхад хангалттай юм.

6. АШИГЛАСАН МАТЕРИАЛ

1. Хуулагч функц –

<https://drive.google.com/file/d/1XnHj4bpb-x0CNJW1eSugWtKwCEc6A19N/view?usp=sharing>

2. Байгуулагч болон устгагч функцууд –

https://drive.google.com/file/d/1JCbme_zvh-7P-7KJ_KfSKw4lAnR2DRuZ/view?usp=sharing

3. Полиморфизм –

https://www.w3schools.com/cpp/cpp_polymorphism.asp

7. ХАВСРАЛТ

```
#include <iostream>
using namespace std;
```

```

/*Ажилчны Классийг Тодорхойлох*/
class Employee{
    private:
        int id;
        char *name;
        char *position;
        float workHour;
        int hourSalary;
        float ceoSalary();
    public:
        void input();
        void output();
        float salary();
        bool addHour(float );
        Employee(){
            name = new char[20];
            position = new char[10];
        }
        Employee(int tsagiinHuls){
            name = new char[20];
            position = new char[10];
            workHour = tsagiinHuls;
        }
        Employee(int tsagiinHuls, int dugaar, char *ner, char
*tushaal, float tsag){
            name = new char[20];
            position = new char[10];
            id = dugaar;
            strcpy(name, ner);
            strcpy(position, tushaal);
            workHour = tsag;
        }
        ~Employee(){
            delete name;
            delete position;
        }
    }

```



```

        cout << "Deleted successfully\n";
    }

};

/*Гараас утга авах функц*/
void Employee::input() {
    cout << "ID: ";
    cin >> id;
    cout << "Ner: ";
    cin >> name;
    cout << "Tushaal: ";
    cin >> position;
    cout << "Ajillasan tsag: ";
    cin >> workHour;
    cout << "Tsagiin tsalin: ";
    cin >> hourSalary;
}

/*Мэдээлэл дэлгэцлэх*/
void Employee::output() {
    cout << "ID: " << id << endl;
    cout << "Ner: " << name << endl;
    cout << "Tushaal: " << position << endl;
    cout << "Ajillasan tsag: " << workHour << endl;
}

/*Нийт цалин бодох*/
float Employee::salary() {
    int total = 0, add;
    cout << "Nemelt tsag: ";
    cin >> add;
    if(addHour(add))
        workHour += add;
    else
        cout << "Buruu utga oruulsan\n";

    if(strcmp(position, "ceo") == 0)

```

```

        total = ceoSalary() + hourSalary*workHour;
    else
        total = hourSalary*workHour;
    return total;
}
/*Захирлын цалин бодох*/
float Employee::ceoSalary(){
    int hour = 20000;
    return hour*workHour;
}
/*Ажилласан цаг нэмэгдүүлэх*/
bool Employee::addHour(float nemelt){
    if(nemelt > 0 && nemelt <= 24){
        return 1;
    }
    return 0;
}

int main(){
    int total[3];
    Employee *ajilchid = new Employee[3];
    for(int i = 0; i < 3; i++){
        (ajilchid+i)->input();
        *(total+i) = (ajilchid+i)->salary();
        cout << endl << "Niit tsalin: " << *(total+i) << endl;
    }

    int temp, j, i;
    for(i = 0; i < 3; i++){
        for(j = i + 1; j < 3; j++){
            if(total[j] < total[i]){
                temp = total[i];
                total[i] = total[j];
                total[j] = temp;
            }
        }
    }
}

```

```
        }  
    }  
}  
  
for(i = 0; i < 3; i++)  
    cout << total[i]<< endl;  
}
```