

# (Лаборатори №6)

ХШУИС, МКУТ

Компьютерийн ухааны хөтөлбөр

Г.Мөнх-Оргил 18b1num2048

## 1. ОРШИЛ/УДИРТГАЛ

Объект хандлагат C хэлний чухал хэсэг болох удамшилтай 8р лабораторийн хүрээнд суралцах ба лабораторийн ажлын хүрээнд удамшилтай холбоотой асуудал, алдаануудтай танилцаж түүнийг шийдэх чадварыг эзэмшэхийг зорино.

## 2. ЗОРИЛГО

1. Удамшил гэж юу вэ ?
2. Удамшилийн горим.
3. Удамшилын давуу тал
4. Удамшилын хэдэн төрөл байдаг вэ?

## 3. ОНОЛЫН СУДАЛГАА

### 3.1 Удамшил гэж юу вэ?

Объект хандлагат програмчлалд удамшил нь нэгэнт бий болсон зүйлийг дахин хэрэглэхтэй холбоотой ойлголт. Байгаа классаас бичигдсэн кодыг дахин ашиглах замаар шинэ класс үүсгэхийг удамшил гэнэ. Өөрөөс нь шинэ класс үүсэж байгааг үндсэн класс, эх класс гэнэ. Шинээр үүсэж байгаа классыг дэд класс, хүүхэд класс гэнэ. Нэгэнт үүссэн классыг өөр програмд аль болох өргөн ашигласанаар хийх ажлын хэмжээ багасахаас гадна алдаа багасан, хугацаа богиносж өртөг буурдаг. Мөн C++ хэлний классын санг ашиглаж шинээр үүсэх классдаа ашиглах нь ОХП-д удамшилтай холбоотой.

### 3.2 Удамшилын горим

Эх классын гишүүн өгөгдөл, гишүүн функцийн хандалтын түвшин хүүхэд класст удамшин очихдоо удамшлын горимоор тодорхойлогдоно.

#### Private горимоор удамших

Эх класс private горимоор удамшихад түүний public гишүүд удамшиж шинээр үүсэх классын private гишүүд болно. Үүнээс удамших классын гишүүд эх классынхаа private гишүүдэд хандаж чадахгүй.

Эх класс private горимоор удамших үед түүний private гишүүд огт удамшихгүй. Ингэхлээр удамших классын гишүүд эх классынхаа private гишүүдэд хандаж чадахгүй.

Эх класс private горимоор удамшихад үед түүний protected хандалтын түвшинтэй гишүүд удамших классын private гишүүд болох ба гишүүн функцууд энэ гишүүд рүү хандах боломжтой байна.

Жишээ нь:

```
class BB
{
    private:
        int aa;
    protected:
        int bb;
    public:
        int cc;
}
class DD : private BB
{
    private:
        int bb;
        int cc;
}
```

### Public горимоор удамших

Эх класс public горимоор удамших тохиолдолд түүний public гишүүд удамшиж үүсэх классын public гишүүд болно. Удамших классын функц ба объект эх классынхаа public гишүүд рүү хандана.

Харин эх классын private гишүүд огт удамшихгүй тул удамших классын гишүүд эх классынхаа private гишүүд рүү хандаж чадахгүй.

Эх класс public горимоор удамших үед түүний protected гишүүд удамших классынхаа protected гишүүд болох тул ийм гишүүд рүү удамших классын функцээс хандаж болно. Жишээ нь

```
class BB
{
    private:
        int aa;
    protected:
        int bb;
    public:
        int cc;
}
class DD : public BB
{
    protected:
        int bb;
    public:
        int cc;
}
```

### Protected горимоор удамших

Эх класс protected горимоор удамших үед түүний public гишүүд удамших классынхаа protected гишүүд болох тул ийм гишүүд рүү удамших классын функцээс хандаж болно.

Харин эх классын private гишүүд огт удамшихгүй тул удамших классын гишүүд эх классынхаа private гишүүд рүү хандаж чадахгүй.

Эх класс protected горимоор удамших үед түүний protected гишүүд удамших классынхаа protected гишүүд болох тул ийм гишүүд рүү удамших классын функцээс хандаж болно.

Жишээ нь:

```
class BB
{
    private:
        int aa;
    protected:
        int bb;
    public:
        int cc;
}
class DD : protected BB
{
    protected:
        int bb;
        int cc;
}
```

### 3.3 Удамшилын давуу талууд

Удамшил нь нэгэнт бий болсон зүйлийг дахин хэрэглэхтэй холбоотой ойлголт юм. Үүнийг эргэн хэрэглэх нь тухайн програмыг хөгжүүлэх цаг хугацааг хэмнэхээс гадна гарч буй алдааг багасгах, шаардлагатай үед дахин ашиглаж болох давуу талуудтай.

### 3.4 Удамшилын 5-н төрөл

#### 1. Нэг → нэг ( дан, энгийн)

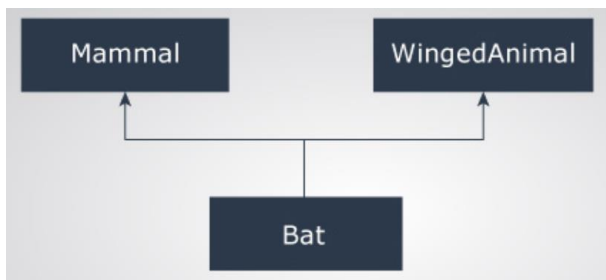
Удаших класс нь зөвхөн нэг эх класстай байна.

Хүн (эх класс) ←←← Оюутан (удамших класс)

```
class person{
};
class student : public person{
}
```

#### 2. Нэг → Олон ( нийлмэл )

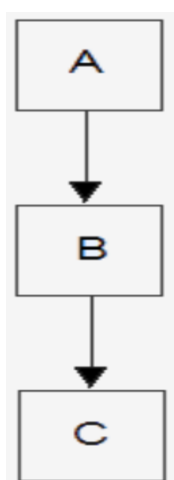
Удамших класс нь хоёроос цөөнгүй эх класстай байж болно.



Багваахай класс нь сүүн тэжээлтэн болон нисдэг амьтадын классаас удамшисан.

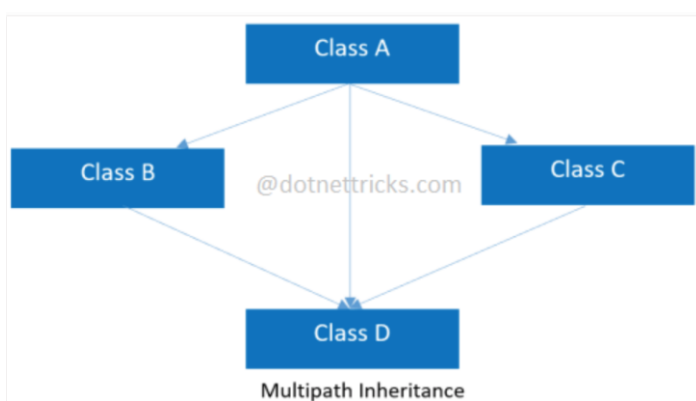
### 3. Олон түвшинт

Удамших класс нь өөрөө удамших класс болж хувирах.



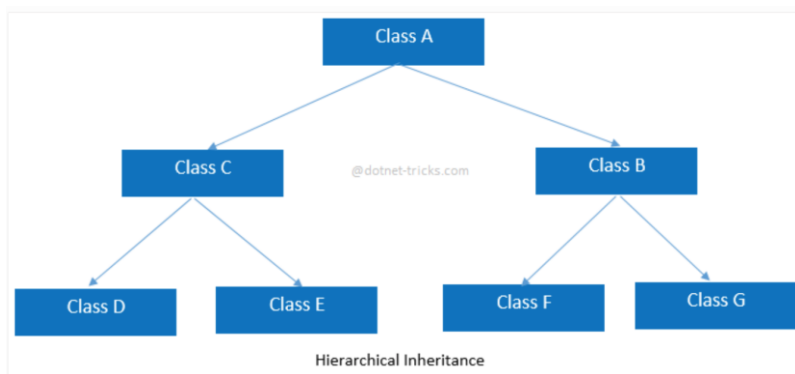
### 4. Шаталсан

Нэг эх классаас хоёроос доошгүй удамших класс үүсэх бол ийм удамшлыг шаталсан удамшил гэнэ.



### 5. Холимог

Холимог удамшил нь олоноос нэг буюу нийлмэл болон шаталсан удамшил хоёрын нийлэмж байна.



<https://www.dotnettricks.com/learn/oops/understanding-inheritance-and-different-types-of-inheritance>

#### 4. ХЭРЭГЖҮҮЛЭЛТ

1. Дүрс класс ашиглан бусад олон хэмжээст дүрсийг удамшуулах ажил.

```
#include <iostream>
#include <math.h>
using namespace std;

class shape
{
protected:
    int x;
    int y;
    char name;
public:
    float findS();
    float perimeter();
    void showdata();
};

class twoDimen:public shape
{
protected:
    int coordinate;
public:
    void showdata();
};
```

```

class circle:public twoDimen, public shape
{
    protected:
        int radius;
    public:
        void showdata();
        float findS();
        float perimeter();
};
float circle::perimeter()
{
    return radius*3.14*2;
}
float circle::findS()
{
    return 3.14*radius*radius;
}
void circle::showdata()
{
    cout<<"\n~~~~~Circle~~~~~\n";
    cout << "toirgiin radius:" << endl;
    cin >> radius;
    cout << "toirgiin urt:" << perimeter() << endl;
    cout << "toirgiin talbai:" << findS() << endl;
}

```

```

class square:public twoDimen, public shape
{
    protected:
        int tal;
    public:
        void showdata();
        float findS();
        float perimeter();
};

float square::perimeter()
{
    return 4*tal;
}

float square::findS()
{
    return tal*tal;
}

void square::showdata()
{
    cout << "\n~~~~~Square~~~~~\n";
    cout << "Kwadratiin tal:"<< endl;
    cin >> tal;
    cout << "Kwadratiin perimeter:" << perimeter() << endl;
    cout << "Kwadratiin talbai:" << findS() << endl;
}

```

```

class triangle:public twoDimen, public shape
{
    protected:
        int tal;
    public:
        void showdata();
        float findS();
        float perimeter();
};
float triangle::perimeter()
{
    float p = tal+tal+tal;
    return p;
}
float triangle::findS()
{
    float p = (tal+tal+tal)/2;
    return sqrt(p*(p-tal)*(p-tal)*(p-tal));
}
void triangle::showdata()
{
    cout << "\n~~~~~Triangle~~~~~\n";
    cout << "Gurwaljnii tal:" << endl;
    cin >> tal;
    cout << "Gurwaljnii perimeter:" << perimeter() << endl;
    cout << "Gurwaljnii talbai:" << findS() << endl;
}

```

```

int main()
{
    circle cir;
    square sq;
    triangle tr;
    cir.showdata();
    sq.showdata();
    tr.showdata();
}

```

## 5.Ашигласан материал

1. <https://www.geeksforgeeks.org/constructors-c/>
2. [https://www.tutorialspoint.com/cplusplus/cpp\\_constructor\\_destructor.html](https://www.tutorialspoint.com/cplusplus/cpp_constructor_destructor.html)
3. <https://www.dotnettricks.com/learn/oops/understanding-inheritance-and-different-types-of-inheritance>



4. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар

## **6. ДҮГНЭЛТ**

Дээрх ажлын хүрээнд объект хандлагат технологийн үндсэн ойлголтуудын нэг болох удамшлын шинж чанарыг ашиглан тойрог, гурвалжин, квадрат гэх мэт дүрсүүдийн классыг байгууллаа. Классуудыг байгуулж байх явцад удамшил ашигласан учир мөр бичлэг хэмнэх, алдаа гарахаас сэргийлэх, дахин давтагдах үйлдлүүдийг нэгтгэн эх классаас удамшуулсан учир програм уншвартай, ажлын хэмжээ болон гарч болох алдаа багассан, цаг хугацаа хэмнэсэн гэх мэт давуу талууд гарлаа.