



Цэрэнбат Номин-Эрдэнэ  
(20B1NUM0426)

## **ICSI201 - Объект хандлагат програмчлал**

МУИС-ХШУИС  
Мэдээллийн технологи

# Лаборатори №7

Цэрэнбат Номин-Эрдэнэ

## 1. ОРШИЛ

Энэ лабораторийн ажлийн хүрээнд Объект хандлагат C хэлний чухал хэсэг болох удамшлыг судалж, тайлбарлаж, ойлгож авна.

## 2. ЗОРИЛГО

- Лабораторийн ажлыг алдаагүй зөв хийж гүйцэтгэх.
- Удамшил болон удамшлын горим тухай уншиж судлах
- Удамшлын давуу талуудыг тоочих
- Удамшлын төрөл бүрийг тайлбарлан бичих

## 3. ОНОЛЫН СУДАЛГАА

### 3.1 Удамшил гэж юу вэ? (тодорхойлолт, C++/java хэл дээр хэрхэн хэрэгжүүлдэг талаар бичнэ)

Удамшил гэдэг нь өмнө тодорхойлогдсон классаас шинэ класс үүсгэх арга юм. Өөрөөр хэлбэл тодорхой нэг классын шинж чанарууд, үйлдлүүдийг өөр нэг класс дотор дахин тодорхойлолгүй шууд авч ашиглах боломжийг удамшил гэх бөгөөд энэ нь эх класс, хүү класс гэсэн ойлголтыг бий болгодог. Шинээр класс үүсгэсэн классыг эх класс харин шинээр үүссэн классыг хүүхэд класс гэдэг. Мөн удамшиж үүсэх класс нь эх классын шинжийг өвлөж авахаас гадна өөрийн гэсэн шинжтэй өөрөөр хэлбэл эх класст байхгүй гишүүдтэй байдаг.

Жишээ нь:

```
class person{
    public:
        char name[20];
        int age ;
};
```

```
class student: public person{
    public:
        int lesson;
        int point;
};
```

Дээрх жишээнд оюутан нэртэй класс нь хүн нэртэй класаас удамших класс бөгөөд эх классынхаа бүх шинжийг өвлөн авнанжтэй өөрөөр хэлбэл эх класст байхгүй гишүүдтэй байдаг.

Удамшсан классаас түүний эх класс руу хандахдаа удамшсан классын байгуулагч функцийн объектоор дамжуулан дуудах үед эх классын байгуулагч функц дуудагддаг

#### **C++ хэлэнд хэрэгжүүлэлт:**

```
class Parent {  
    public:  
        int id_p;  
};  
class Child : public Parent {  
    public:  
        int id_c;  
};
```

Дээрх жишээнд Child класс нь Parent нэртэй класаас удамших бөгөөд эх классынхаа бүх шинжийг өвлөн авна

#### **Java- хэлэнд хэрэгжүүлэлт:**

JAVA-ийн класс бүр ямар нэгэн классыг удамшуулсан байдаг. Удамшуулахдаа extends түлхүүр үг ашиглана.

```
class Teacher {  
    String designation = "Teacher";  
    ...  
}  
  
public class PhysicsTeacher extends Teacher {  
    String mainSubject = "Physics";  
    ...  
}
```

### 3.2 Удамшлын горим. public, private, protected горимын талаар тайлбарлаж жишээгээр батална.

Удамшилын горим гэдэг нь эх класс хэрхэн удамшихыг заадаг ойлголт юм. Өөрөөр хэлбэл Удамшлын горим нь эх классаас өвлөглөж байгаа гишүүн өгөгдөл болон гишүүн функцийн хандалтын түвшинг тодорхойлдог.

Хэрэв удамшлын горимыг зааж өгөөгүй бол анхнаасаа private горимтой байдаг.

Удамшилын горимын 3 төрөл байдаг.

1. public
2. private
3. protected

**Private** горимоор удамшвал эх классын protected болон public түвшинтэй гишүүн өгөгдөл, гишүүн функц шинээр үүсэх хүүхэд классд private хандалтын түвшинтэй болж өвлөгдөнө.

**Public** горимоор удамшвал өвлөгдөж буй гишүүн өгөгдлийн горим өөрчлөгдөхгүй. Мөн эх классын private гишүүд огт удамшихгүй.

### 3.3 Удамшлын давуу талуудыг тоочин бичиж бодит жишээн дээр тайлбарла.

Удамшил нь нэгэнт бий болсон зүйлийг дахин хэрэглэхтэй холбоотой ойлголт юм.

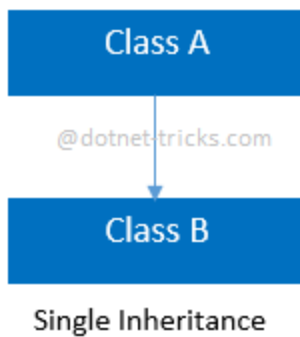
Удамших ба удамшуулах үйлдлүүд нь дараах давуу талуудтай.

- Ихэнх програмын хувьд эх классынхаа шинжийг өвлөх олон класс хэрэгтэй бөгөөд тэдгээр эх классыг нэг эх классаас үүсгэж эх классыг өгөгдлийн ерөнхий бүтэц хэлбэрээр тодорхойлох боломжтой
- Удамшилыг ашигласнаар програмыг хөгжүүлэх цаг хугацааг хэмнэхээс гадна програмд гарч болох алдааг багасган програм боловсруулагдах хугацаа богиносно.

### 3.4 Удамшлын хэдэн төрөл байдаг вэ? Тус бүрийг тайлбарлан бич.

#### 1. Нэг - нэг ( дан, энгийн)

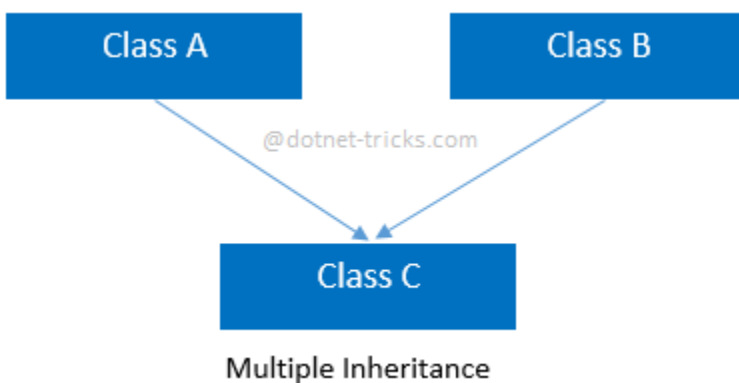
Удаших класс нь зөвхөн нэг эх класстай байна.



## 2. Нэг - Олон ( нийлмэл )

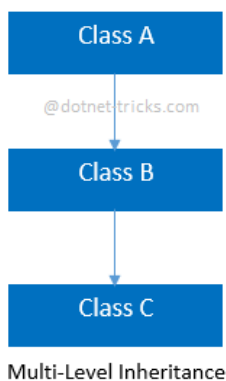
Удамших класс нь хоёроос цөөнгүй эх класстай байж болно.

Жишээ нь аав ээж хоёроос хүү удамшина гэвэл аав ээж 2 нь эх классууд болно



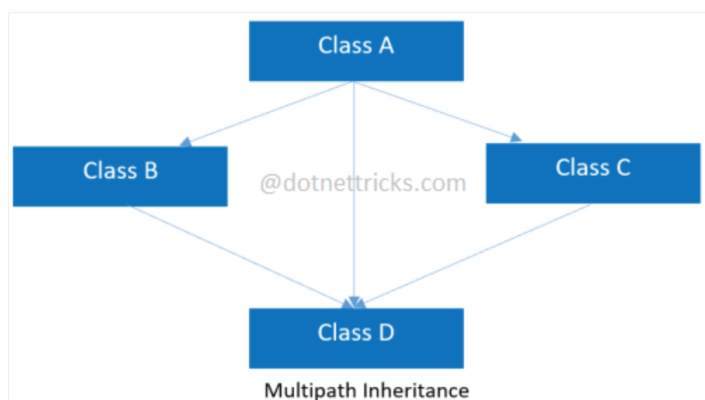
## 3. Олон түвшинт

Удамших класс нь өөрөө удамших класс болж хувирах.



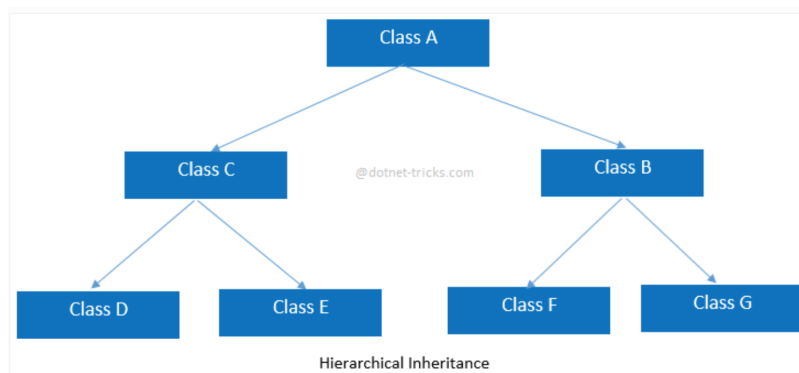
#### 4. Шаталсан

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



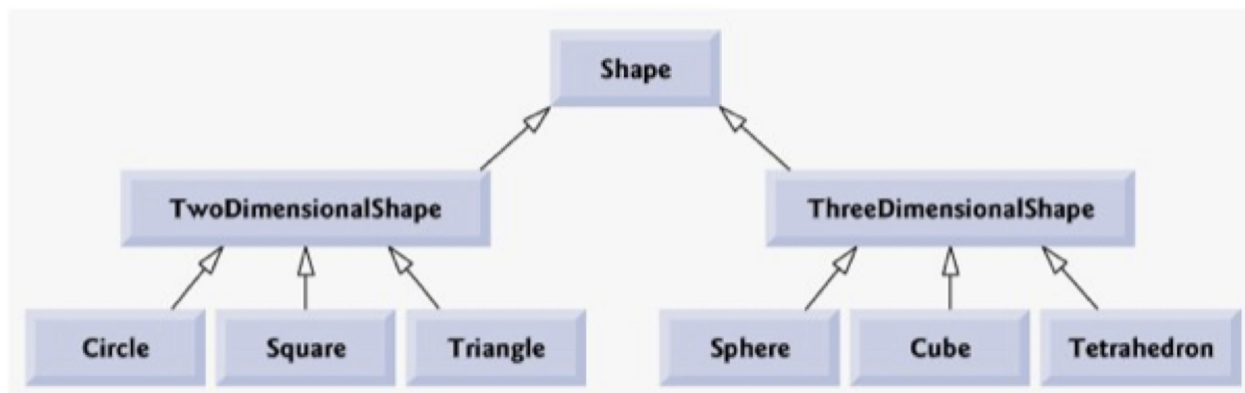
#### 5. Холимог

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



#### 4. ХЭРЭГЖҮҮЛЭЛТ

Зураг 1-д дүрсэлсэн удамшлын модны дагуу хоёр хэмжээст геометрийн дүрсүүдийн классыг байгуул.



CLASS SHAPE:

```
lab7 test1.cpp
1  #include<iostream>
2  #include<string.h>
3  #include<math.h>
4  #define pi 3.14
5  using namespace std;
6
7  class shape{
8
9      protected:
10         char *name;
11
12     public:
13         shape();
14         ~shape();
15
16 };
17
```

CLASS TWO DIMENSIONAL SHAPE:

```
28
29 class TwoDShape : public shape{
30
31     protected:
32         int *x;
33         int *y;
34
35     public:
36         float area();
37         float primetr();
38
39 };
```

CLASS TOIROG:

```
42
43 class toirog: public TwoDShape{
44     private:
45         int r;
46
47     public:
48         float area();
49         float urt();
50
51     toirog(){
52         name = new char[7];
53         strcpy(name, "toirog");
54         x = new int[1];
55         y = new int[1];
56         r = 0;
57     }
58
59     ~toirog(){
60         delete name;
61         delete x;
62         delete y;
63     }
64 }
```



CLASS ZUV GURVALJIN:

```
153
154 = class Zgurvaljin: public TwoDShape{
155     private:
156         float a;
157     public:
158         float area();
159         float primetr();
160
161
162 =     Zgurvaljin(){
163         name = new char[9];
164         strcpy(name, "ZuwGurvaljin");
165         x = new int[3];
166         y = new int[3];
167     }
168
169 =     ~Zgurvaljin(){
170         delete name;
171         delete x;
172         delete y;
173     }
174
```

CLASS KVADRAT:

```
92
93 = class kvadrat: public TwoDShape{
94     private:
95         float a;
96
97     public:
98         double area();
99         double primetr();
100
101 =     kvadrat(){
102         name = new char[7];
103         strcpy(name, "kvadrat");
104         x = new int[4];
105         y = new int[4];
106         a=0;
107     }
108
109 =     ~kvadrat(){
110         delete name;
111         delete x;
112         delete y;
113     }
114
```

C:\Users\nomin\OneDrive\Documents\oop\lab7 test1.exe

Toirog coordinat x: 5

x ni 5 bolloo.

Toirog coordinat y: 2

y ni 2 bolloo.

Toiognii radius:3

Toiognii area:28.26

Toingnii urt:18.84

Kvadrat a:5

Kvadratnii coordinat x:3

x ni 3 bolloo.

x1 ni 8 bolloo.

x2 ni 8 bolloo.

x3 ni 3 bolloo.

Kvadratnii coordinat y:4

y ni 4 bolloo.

y1 ni 4 bolloo.

y2 ni -1 bolloo.

y3 ni -1 bolloo.

Kvadratnii area:25

Kvadratnii primetr:20

Gurvaljnii Tal:5

Gurvaljnii tal ni5bolloo.

Zuv Gurvaljnii coordinat x:3

x ni 3 bolloo.

x1 ni 8 bolloo.

x2 ni 3 bolloo.

Zuv Gurvaljnii coordinat y:4

y0 ni 1442128 bolloo.

y1 ni 0 bolloo.

y2 ni 1442128 bolloo.

Zuv Gurvaljnii area:10.8253

Zuv Gurvaljnii primetr:15

-----  
Process exited after 22.97 seconds with return value 3221226356

Press any key to continue . . .

## 5. XABCPAJIT

```
#include<iostream>
#include<string.h>
#include<math.h>
#define pi 3.14
using namespace std;

class shape{

protected:
    char *name;

public:
    shape();
    ~shape();

};

shape::shape(){
    name = new char[2];
    strcpy(name, "");
}

shape::~~shape(){
    delete name;
}

// _____ 2DSHAPE _____ //

class TwoDShape : public shape{

protected:
    int *x;
    int *y;

public:
    float area();
    float primetr();
}
```

```

};

// _____ TOIROG _____ //

class toirog: public TwoDShape{
private:
    int r;

public:
    float area();
    float urt();

    toirog(){
        name = new char[7];
        strcpy(name, "toirog");
        x = new int[1];
        y = new int[1];
        r = 0;
    }

    ~toirog(){
        delete name;
        delete x;
        delete y;
    }

    void set_x(int _x){
        x=&_x;
        cout<<"x ni "<<*x<<" bolloo.\n";
    }

    void set_y(int _y){
        y=&_y;
        cout<<"y ni "<<*y<<" bolloo.\n";
    }

    void set_r(int _r){
        r=_r;
    }
}

```

```

};

float toirog::area(){
    return pi * (r * r);
}

float toirog::urt(){
    return 2*pi*r;
}

// _____ KVADRAT _____ //

class kvadrat: public TwoDShape{
private:
    float a;

public:
    double area();
    double primetr();

    kvadrat(){
        name = new char[7];
        strcpy(name,"kvadrat");
        x = new int[4];
        y = new int[4];
        a=0;
    }

    ~kvadrat(){
        delete name;
        delete x;
        delete y;
    }

    void set_a(float _a){
        a=_a;
    }
}

```

```

        void set_x(int _x){
            x[0] = _x;
            cout<<"x ni "<<x[0]<<" bolloo.\n";
            x[1] = _x + a;
            cout<<"x1 ni "<<x[1]<<" bolloo.\n";
            x[2] = _x + a;
            cout<<"x2 ni "<<x[2]<<" bolloo.\n";
            x[3] = _x;
            cout<<"x3 ni "<<x[3]<<" bolloo.\n";
        }

        void set_y(int _y){
            y[0] = _y;
            cout<<"y ni "<<y[0]<<" bolloo.\n";
            y[1] = _y;
            cout<<"y1 ni "<<y[1]<<" bolloo.\n";
            y[2] = y[3] = _y - a;
            cout<<"y2 ni "<<y[2]<<" bolloo.\n";
            cout<<"y3 ni "<<y[3]<<" bolloo.\n";
        }

};

double kvadrat::area(){
    return a * a;
}

double kvadrat::primetr(){
    return a*4;
}

// _____ GURVALJIN _____ //

class Zgurvaljin: public TwoDShape{
private:
    float a;
public:
    float area();

```

```
float primetr();
```

```
Zgurvaljin(){  
    name = new char[9];  
    strcpy(name, "ZuwGurvaljin");  
    x = new int[3];  
    y = new int[3];  
}
```

```
~Zgurvaljin(){  
    delete name;  
    delete x;  
    delete y;  
}
```

```
void set_x(int _x){  
    x[0] = _x;  
    cout<<"x ni "<<x[0]<<" bolloo.\n";  
    x[1] = _x + a;  
    cout<<"x1 ni "<<x[1]<<" bolloo.\n";  
    x[2] = _x;  
    cout<<"x2 ni "<<x[2]<<" bolloo.\n";  
}
```

```
void set_y(int _y){  
    x[0] = _y;  
    cout<<"y0 ni "<<y[0]<<" bolloo.\n";  
    x[1] = _y - a;  
    cout<<"y1 ni "<<y[1]<<" bolloo.\n";  
    x[2] = _y - a;  
    cout<<"y2 ni "<<y[2]<<" bolloo.\n";  
}
```

```
void set_a(float _a){  
    a=_a;  
    cout<<"Gurvaljnii tal ni"<<a<<"bolloo.\n";  
}
```

```
};
```

```

float Zgurvaljin::area(){
    float h = sqrt(a*a-a/2*a/2);
    return h*a/2;
}

float Zgurvaljin::primetr(){
    return a*3;
}

int main(){
    toirog t;
    kvadrat k;
    Zgurvaljin zg;

    int r,tx,ty,kx,ky,gx,gy;
    float ka,ga;
    cout<<"Toirog coordinat x: ";
    cin>>tx;
    t.set_x(tx);
    cout<<"Toirog coordinat y: ";
    cin>>ty;
    t.set_y(ty);
    cout<<"Toirognii radius:";
    cin>>r;
    t.set_r(r);
    cout<<"Toirognii area:"<<t.area()<<endl;
    cout<<"Toirgnii urt:"<<t.urt()<<endl;

    cout<<"\nKvadrat a:";
    cin>>ka;
    k.set_a(ka);
    cout<<"Kvadratnii coordinat x:";
    cin>>kx;
    k.set_x(kx);
    cout<<"Kvadratnii coordinat y:";
    cin>>ky;
    k.set_y(ky);
}

```



```
cout<<"\nKvadratnii area:"<<k.area()<<endl;
cout<<"Kvadratnii primetr:"<<k.primetr()<<endl;

cout<<"\nGurvaljnii Tal:";
cin>>ga;
zg.set_a(ga);
cout<<"Zuv Gurvaljnii koordinat x:";
cin>>gx;
zg.set_x(gx);
cout<<"Zuv Gurvaljnii koordinat y:";
cin>>gy;
zg.set_y(gy);

cout<<"\nZuv Gurvaljnii area:"<<zg.area()<<endl;
cout<<"Zuv Gurvaljnii primetr:"<<zg.primetr()<<endl;
}
```