

(Лаборатори №2)

Э.Багабанди

ХШУИС - ийн 3-р түвшиний оюутан, 19B1NUM0700@num.edu.mn

1. ОРШИЛ/УДИРТГАЛ

Уг лабораторийн хүрээнд өгөгдсөн бодлогуудыг бодож, тайлбарлахад ашиглагдсан C++ хэлний онолын ойлголт(хаяг, хаяган хувьсагч, заалтан хувьсагч, хэрэглэгчийн функц тодорхойлох, хүснэгтэд хаягаар хандах) - ыг судалж, эзэмшсэн болно.

2. ЗОРИЛГО

Энэхүү лабораторийн ажлын хүрээнд C++ хэлний хаяг, хаяган хувьсагч, заалтан хувьсагч, түүний хэрэглээг ойлгох, хэрэглэгчийн функцийг заалтан болон хаяган хувьсагч төрлийн параметртэйгээр тодорхойлох, хүснэгтийг хаягаар нь хандаж боловсруулах онолын ойлголтуудыг судалж, үүнийг ашиглан лабораторт өгөгдсөн бодлогуудыг бодож, гарч буй үр дүнг тайлбарлана.

3. ОНОЛЫН СУДАЛГАА

3.1 Хаяг, хаяган болон заалтан хувьсагч

- **Хаяг**
 - Хаяг бол зарлагдсан хувьсагчийг санах ойд хаана байрлаж байгааг заах тоон утга юм.
- **Хаяган хувьсагч**
 - Хаяг нь тоон утга тул түүнийг хувьсагч руу хадгалж болох бөгөөд хаяг хадгалах хувьсагчийг хаяган хувьсагч гэнэ.
- **Заалтан хувьсагч**
 - Си хэлэнд байдаггүй, зөвхөн C++ хэлний шинэ боломжийн нэг юм. Заалтан хувьсагчийг тодорхой нэр бүхий хувьсагчтай холбож үүсгэх бөгөөд хэрэв `x_ref` нь `x` хувьсагчийн заалтан нэр бол `x` ба `x_ref` нь нэг ойн хоёр өөр нэр болох учир эдгээр нэрийн алийг хэрэглэж болно. Заалтан хувьсагчийн функцийн параметрээр хэрэглэхэд нэн тохиромжтой ба ингэснээр ой хэмнэхээс гадна эх өгөгдөл рүү нь шууд хандан бодолт хийх боломжтой болно.
- **Зарлах, тодорхойлох**
 - Хаяг: `int a;` \Rightarrow тухайн хувьсагчийн хаягийг `'&a'` гэж буюу `&` оператор ашиглан авна.
 - Хаяган хувьсагч: `int a;` \Rightarrow `int &a;` хаяган хувьсагчийг тодорхойлохдоо түүнд ямар утга төрлийн хувьсагчийн хаяг хадгалах гэж байгааг зааж өгдөг. Учир нь бүх төрлийн хувьсагч ойн хаягийн хувьд ижил боловч тэдгээрт шаардагдах ой нь тоо

хэмжээ болон компьютерийн дотоод дүрслэлий хувьд хоорондоо ялгаатай.

- Заалтан хувьсагч: Хувьсагч, заалтан хувьсагчийг / **int x; => int &x_ref = x;** гэж холбоно.

4. ХЭРЭГЖҮҮЛЭЛТ

1. Санах ойн хаяг гэж юу вэ? Хаягийн хэмжээ хэд байдаг вэ?

- ☐ Санах ойн хаяг гэдэг нь зарлагдсан хувьсагчийг санах ойд хаана байрлаж байгааг заах тоон утга. Санах ойн хаяг нь том тоо учир өөр дээр байрлах утгын төрлөө үл хамаарч 8 байт хэмжээтэй байна.

2. Доорх кодыг туршиж үр дүнг тайлабарлан бич.

- ☐ **char *p1;** // char төрлийн заагч хувьсагч зарласан.
- ☐ **int *p2;** // int төрлийн заагч хувьсагч зарласан.
- ☐ **double *p3;** // double төрлийн заагч хувьсагч зарласан.
- ☐ **cout<<sizeof(p1)<<sizeof(p2)<<sizeof(p3);** // дээр зарласан 3н төрлийн заагч хувьсагчийн санах ойд эзлэх хэмжээг хэвлэж байна.
- ☐ Ойд эзлэх хэмжээг хэвлэж байна. Бүгд ижилхэн 8 byte гэж гарсан. Учир нь тухайн хувьсагчууд нь бүгд заагч хувьсагч тул ямар нэг хувьсагчийн хаягийг хадгалах бөгөөд харин төрөл нь зөвхөн тухайн заагч хувьсагчийн ямар төрлийн хувьсагчийн хаягийг хадгалахыг зааж өгдөг. Иймд тухайн хувьсагчид хадгалах хаяг нь 8 byte - ийн хэмжээтэй тоон утга байна.

3. Доорх кодыг туршиж мөр бүрийн үр дүнг тайлбарлан бич.

- ☐ **int a=125;** // int төрлийн гарааны утга нь 125 байх, а гэх хувьсагч зарласан.
- ☐ **int *p = &a;** //int төрлийн p хаяган хувьсагч зарлаж, түүнд а хувьсагчийн хаягийг & операторын тусламжтайгаар гарааны утга болгон олгосон.
- ☐ **cout<<p;** //p хувьсагчийн утгыг буюу а хувьсагчийн хаягийг хэвлэсэн.
- ☐ **cout<<*p;** //p хувьсагчид хадгалагдах хаяг дээрх утгыг буюу а хувьсагчийн утгыг хэвлэсэн.
- ☐ **p++;** // p хувьсагчийн утгыг буюу а хувьсагчийн хаягийг 4 byte/int/ - аар нэмэгдүүлсэн.
- ☐ **cout<<p;** // p хувьсагчийн утгыг буюу &a + 4 хэвлэсэн.
- ☐ **cout<<*p;** // p хувьсагчид хадгалагдах хаяг дээрх утгыг буюу &a + 4 хаяг дээр утгыг хэвлэсэн.

4. Доорх кодын мөр бүрийг тайлбарла.

- ☐ **int numbers[5];** //numbers нэртэй int төрлийн 5 урттай хүснэгт зарласан.
- ☒ **int * p;** //бүхэл тоо төрлийн заагч хувьсагч зарласан.
- ☒ **p = numbers;** //p хаяган хувьсагчид numbers хүснэгтийн хамгийн эхний элементийн хаягийг олгосон. Хүснэгтий элементүүдийн хаяг нь санах ойд дарааллан байрлах бөгөөд int төрлийн хувьсагч тул хаяг нь 4, 4н байтын зайтай. Иймд эхний элементийн хаягаар бүх элементэд хандах боломжтой.
- ☒ ***p = 10;** //p хувьсагчид хадгалагдах хаяг дээрх утгыг буюу numbers хүснэгтийн эхний элементийн утгыг 10 болгосон.

- ❑ **p++;** //p хувьсагчийн утга буюу numbers хүснэгтийн хамгийн эхний элементийн хаягийг 4 byte/int/ - аар нэмэгдүүлнэ. Иймд p хаяган хувьсагч нь numbers хүснэгтийн дараагийн элементийн хаягийг заана. Ө.х numbers[1] - ийн хаягийг заана.
 - ❑ ***p = 20;** //p - д numbers[1] хаяг хадгалагдах тул numbers[1] = 20 болно.
 - ❑ **p = &numbers[2];** // p - д шууд & оператор ашиглан numbers[2] - ны хаягийг олгож байна.
 - ❑ ***p = 30;** //p-д numbers[2] хаяг хадгалагдах тул numbers[2] = 30 болно
 - ❑ **p = numbers + 3;** //numbers нь numbers хүснэгтийн хамгийн эхний элементийн хаягийг хадгалах тухайн хаягийг 3 - аар нэмэгдүүлж p хадгалснаар p - д numbers[3] хаяг хадгалагдах юм. Учир 1 int -> 4 byte, 3 int -> 12 byte болох юм.
 - ❑ ***p = 40;** //p - д numbers[1] хаяг хадгалагдах тул numbers[3] = 40 болно.
 - ❑ **p = numbers;** // p - д numbers хүснэгтийн хамгийн эхний элементийн хаяг олгогдоно.
 - ❑ ***(p+4) = 50;** // p - д numbers[0] хаяг байх бөгөөд p + 4 нь numbers[4] хаяг болно. Мөн * оператороор numbers[4] утгыг 50 болгосон.
 - ❑ **for (int n=0; n<5; n++)** //int n нь [0, 4] гүйх for давталт үүсгэсэн
 - ❑ **cout << numbers[n] << ", ";** //давталтын бие дотор numbers[n] - ийг буюу numbers хүснэгтийн бүх элементийг хэвлэж байна.
5. Хаяган хувьсагч ашиглан (функцин параметер нь хаяган хувьсагч байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич.
- ❑ **void swapped(int *x, int *y){** //Заагч параметруудтай функц учир хаяг дамжуулж өгөх шаардлагатай. Учир нь заагч нь хаяг хадгалдаг.
 - ❑ **int tmp = *x;**
 - ❑ ***x = *y;**
 - ❑ ***y = tmp;**
 - ❑ **}**
6. Заалтан хувьсагч (функцин параметер нь заалт байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич.
- ❑ **void swapped(int &x, int &y){** // Дамжиж ирсэн хувьсагчуудын хаягийг &x, &y гэх заалтан хувьсагчуудад хадгалж авна.
 - ❑ **int tmp = x;**
 - ❑ **x = y;**
 - ❑ **y = x;**
 - ❑ **}**
 - ❑

5. ДҮГНЭЛТ

Энэхүү лабораторын ажлын асуултуудын хүрээнд хүснэгт, заагч, хэрэглэгчийн функц гэх ойлголтуудыг ашиглан асуулт, бодлогуудыг шийдвэрлэсэн бөгөөд 5, 6 - аар бодлого дах хэрэглэгчийн функцуудын параметрыг харгалзан заалтан болон хаяган хувьсагчаар авсан. 6 - р бодлогын функцийг дуудаж параметруудыг дамжуулахдаа 2

хувьсагчыг шууд дамжуулж **/swapped(a, b);**/ өгсөн бөгөөд тухайн функц нь дамжиж очсон хувьсагчуудын утгыг заалтан хувьсагчид хадгалан авч түүн дээр боловсруулалт хийнэ. Ингэснээр ой хэмнэх бөгөөд жинхэнэ параметруудийн утга буюу эх өгөгдөл рүү шууд хандан боловсруулалт хийх боломжтой болох юм. Харин 5 - р бодлогын функцыг дуудаж параметруудыг дамжуулахдаа 2 хувьсагчийн хаягийг дамжуулсан **/swapped(&a, &b)/**. Учир нь уг функц нь заагч буюу хаяган хувьсагч бүхий хийсвэр параметруудтэй бөгөөд хаяган хувьсагч нь хаяг хадгалдаг. Иймд тэдгээр хувьсагчуудын хаягийг тэдгээр дээр боловсруулалт хийх боломжтой болно.

6. АШИГЛАСАН МАТЕРИАЛ

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

7. ХАВСРАЛТ

Кодыг энд оруулна.

Бодлого №5:

```
1. /* 5. хаяган хувьсагч ашиглан (функцын параметер нь хаяган хувьсагч байна) хоёр
   хувьсагчийн утгыг солих хэрэглэгчийн функц бич. */
2. #include<iostream>
3. using namespace std;
4.
5. void swapped(int *x, int *y){
6.     int tmp = *x;
7.     *x = *y;
8.     *y = tmp;
9. }
10.
11. int main(){
12.     int a, b;
13.     cout<<"Enter value of a and b: ";
14.     cin>>a>>b;
15.     cout<<"Solihooos umnuh utguud "<<endl;
16.     cout<<"a - iin utga: "<<a<<endl<<"b - iin utga: "<<b<<endl;
17.     swapped(a, b);
18.     cout<<"Solisonii daraah utguud "<<endl;
19.     cout<<"a - iin utga: "<<a<<endl<<"b - iin utga: "<<b;
20.     return 0;
21. }
```

Бодлого №6:

```
1. /* 6. заалтан хувьсагч (функцын параметер нь заалт байна) хоёр хувьсагчийн утгыг
   солих хэрэглэгчийн функц бич. */
```

```

2. #include<iostream>
3. using namespace std;
4.
5. void swapped(int &x, int &y){
6.     int tmp = x;
7.     x = y;
8.     y = tmp;
9. }
10.
11.
12. int main(){
13.     int a, b;
14.     cout<<"Enter value of a and b: ";
15.     cin>>a>>b;
16.     cout<<"Solihooos umnuh utguud "<<endl;
17.     cout<<"a - iin utga: "<<a<<endl<<"b - iin utga: "<<b<<endl;
18.     swapped(&a, &b);
19.     cout<<"Solisonii daraah utguud "<<endl;
20.     cout<<"a - iin utga: "<<a<<endl<<"b - iin utga: "<<b;
21.     return 0;
22. }

```