



Цэрэнбат Номин-Эрдэнэ
(20B1NUM0426)

ICSI201 - Объект хандлагат програмчлал

МУИС-ХШУИС
Мэдээллийн технологи

Лаборатори №2

Цэрэнбат Номин-Эрдэнэ

1. ОРШИЛ

Энэ лабораторийн ажил дээр C++ хэл дээрх санах ойн хаяг, хаяган хувьсагч, заалтан хувьсагчийг судалж, кодыг туршиж, тайлбарлаж, ойлгож авна.

2. ЗОРИЛГО

- Лабораторын ажлыг алдаагүй зөв хийж гүйцэтгэх.
- Санах ойн хаяг болон хаяган хувьсагч, заалтан хувьсагч, дотоод мөр функц, ойн цоорхойн тухай уншиж судлах.
- C++ хэл дээр өгөгдсөн кодуудыг туршиж үр дүнг ба кодын мөр бүрийг шинжлэн тайлбарлан бичих.
- Хаяган болон заалтан хувьсагч ашиглан функц бичих.

3. ОНОЛЫН СУДАЛГАА

3.1 Санах ойн хаяг гэж юу вэ?

Санах ойн хаяг гэдэг нь яг л бидний хүрээлэн буй орчны хаягуудтай ижил ойлголт юм. Санах ойд хаяг олгосноор эдгээр хаягаар дамжуулж санах ойн байт бүрд хандаж болдог.

3.2 Хаяган хувьсагч гэж юу вэ? Хаяган хувьсагчийн хэмжээ хэдэн байт байдаг вэ?

Хаяг бол тоон утга тул түүнийг хувьсагч руу хадгалж болох ба хаяг хадгалах хувьсагчийг хаяган хувьсагч гэнэ. Хаяган хувьсагчийг зарлахдаа түүний нэрний өмнө(*) тавьдаг. Мөн хаяган хувьсагчийн зааж байгаа утганд хандахдаа (*) –г хэрэглэдэг.

Хаяган хувьсагч нь тогтмол unsigned int буюу 32 биттэй бол 4 байт , 64 биттэй бол 8 байт хэмжээтэй байдаг.

3.3 Заалтан хувьсагч гэж юу вэ? Ямар давуу талтай вэ?

Хувьсагчийн нэрний өмнө (&) тэмдэг тавьснаар түний санах ойд байрлах хаягийг тодорхойлох ба үүнийг заалтан хувьсагч гэнэ. Заалтан хувьсагчийг зарлахдаа заавал ямар нэг хувьсагч зааж зарладаг. Мөн заалтан хувьсагч хэзээч тогтмол утгыг зааж чадаггүй. Заалтан хувьсагчийг функцийн параметрээр хэрэглэхэд нэн тохиромжтой байдаг. Ингэснээр ой хэмнэхээс гадна эх өгөгдөл рүү нь шууд хандан бодолт хийх боломжтой болно.

3.4 Дотоод мөр функц гэж юу вэ? Яагаад үүнийг хэрэглэх шаардлагатай байдаг вэ?

Си хэлний inline функц бол програмын хурдыг нэмэгдүүлэх зориулалтаар бүтээгдсэн боломж юм. Дотоод мөр функц ба ерийн Си функц хооронд байх үндсэн ялгаа нь тэдгээрийг програмд хэрхэн холбож өгдөгт оршидог.

3.5 Ойн цоорхой гэж юу вэ? Ямар тохиолдолд үүсдэг вэ? Жишээгээр тайлбарла.

Ойн цоорхой бол new, delete операторуудыг дэс дараатайгаар олон удаа хэрэглэх програмд гарч болох ойлголт юм.

New эсвэл Malloc –оор нөөцөлсөн санах ойг чөлөөлгүй, хувьсагчийг нь орхисон тохиолдолд ахин хандах боломжгүй ой үүсдэг.

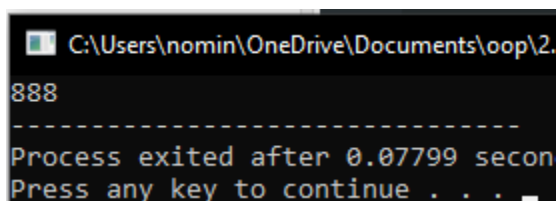
Жишээ:

```
int main(){
int * p = new int;
delete p;
//ойн цоорхой
int * a = new int;
// delete байхгүй тул ойн цоорхой үүснэ
}
```

4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Доорх кодыг туршиж үр дүнг шинжлэн тайлабарлан бич.

```
char *p1;
int *p2;
double *p3;
cout<<sizeof(p1)<<sizeof(p2)<<sizeof(p3); //хэд хэд гэсэн утга хэвлэж байна вэ? Яагаад?
```



Sizeof үйлдлээр тухайн хувьсагчийн утгыг санах ойд хадгалахад хичнээн byte шаардагдажбайгааг тодорхойлдог. Хаяган хувьсагч нь тогтмол unsigned int байдаг учир ижил 8 байт гарна.

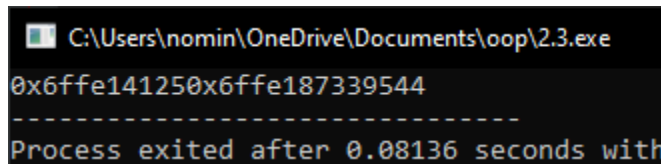
4.2 Доорх кодыг туршиж мөр бүрийн үр дүнг шинжлэн тайлбарла.

```
int a=125; //int төрлийн хувьсагч зарлаж 125 гэсэн утгыг оноож байна
```

```

int *p = &a; //хаяган хувьсагчид а хувьсагчийн хаягийг хадгалсан
cout<<p;    //p-г дэлгэцэнд хэвлэн харуулна
cout<<*p;   // (*) нь массивын эхний элементийг заана
p++;       //p-г нэгээр нэмэгдүүлнэ
cout<<p;    // p-г дэлгэцэнд хэвлэнэ
cout<<*p;   //p- массивын эхний элементийг хэвлэнэ

```



```

C:\Users\nomin\OneDrive\Documents\oop\2.3.exe
0x6ffe141250x6ffe187339544
-----
Process exited after 0.08136 seconds with

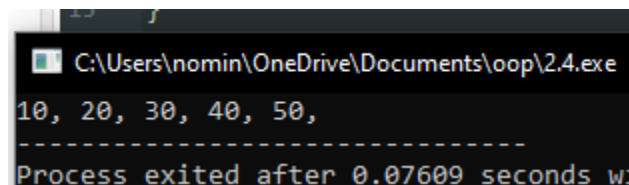
```

4.3 Доорх кодын мөр бүрийг тайлбарла.

```

int numbers[5];           //массив зарлаж байна
int * p;                  //бүхэл тоон төрлийн заагч зарласан
p = numbers; *p = 10;     //p-г тоонуудтай харьцуулаад, p- ийн заасан хаягаар 10-ыг
хуулна
p++; *p = 20;             //p- г нэгээр нэмэгдүүлээд, p-ийн заасан хаягаар 20-г хуулна
p = &numbers[2]; *p = 30; //(p- массивын 3-р элементийг заана, дараа нь 3-р
элементийг 30 болгоно
p = numbers + 3; *p = 40; //массивын 4-р элементийг заана, дараа нь 4-р элементийг
40 болгоно
p = numbers; *(p+4) = 50; //P нь массивын эхний элементийг заана, дараа нь 5-р
элементийг 50 болгоно
for (int n=0; n<5; n++)   //n- д 0 гэж утга оноож, n-нь 5 аас их үед нэгээр
нэмэгдүүлнэ
cout << numbers[n] << ", "; //n хүснэгтээс тоонуудыг хэвлэн дэлгэцэнд харуулна

```



```

C:\Users\nomin\OneDrive\Documents\oop\2.4.exe
10, 20, 30, 40, 50,
-----
Process exited after 0.07609 seconds wi

```

4.4 Хаяган хувьсагч ашиглан (функцын параметер нь хаяган хувьсагч байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич.

```
5.cpp
1  #include<iostream>
2  using namespace std;
3
4  int swap(int *n1, int *n2);
5
6  int main(){
7      int n1 = 1, n2 = 2;
8      cout<<n1<<","<<n2<<"\n";
9      swap(&n1, &n2);
10     cout<<"utgiig solison ur dun: "<<n1<<","<<n2;
11 }
12
13
14 int swap(int *n1, int *n2){
15     int x;
16     x = *n1;
17     *n1 = *n2;
18     *n2 = x;
19 }
```

```
C:\Users\nomin\OneDrive\Documents\oop\5.exe
1,2
utgiig solison ur dun: 2,1
-----
Process exited after 0.07513 seconds with
Press any key to continue . . .
```

4.5 Заалтан параметр (функцын параметер нь заалт байна) ашиглан хоёр хувьсагчийн байрыг солих хэрэглэгчийн функц бич.

```
5.cpp 7.cpp
1  #include<iostream>
2  using namespace std;
3
4  int swap(int &n1, int &n2);
5
6  int main(){
7      int n1 = 1, n2 = 2;
8      cout<<n1<<","<<n2<<"\n";
9      swap(n1, n2);
10     cout<<"utgiig solison ur dun: "<<n1<<","<<n2;
11 }
12
13
14 int swap(int &n1, int &n2){
15     int x;
16     x = n1;
17     n1 = n2;
18     n2 = x;
19 }
```

```
C:\Users\nomin\OneDrive\Documents\oop\7.exe
1,2
utgiig solison ur dun: 2,1
-----
Process exited after 0.07265 seconds with
Press any key to continue . . .
```

