Лабораторн 5:

1. Байгуулагч функц хэзээ дуудагддаг вэ?

Байгуулагч функц нь обьект шинээр үүсэх үед түүнд зориулан санах ой бэлдэхээр анхдагч эсвэл хэрэглэгчийн тодорхойлсон байгуулагч функц дуудагддаг. Нэг объект байгуулах үед нэг л удаа дуудагдана.

2. Устгагч функц хэзээ дуудагдах вэ?

Устгагч функц нь тухайн классын обьектын дуудсан функцын үйлчлэх хүрээ дуусгавар болоход дуудагдаж ажиллана.

3. Хуулагч байгуулагч гэж юу вэ? Ач холбогдол нь юу вэ?

Хуулагч байгуулагч гэдэг нь шинэ обьект болон түүний өмнөх аль нэгэн обьектын утгыг хуулбарлаж авах боломжийг олгодог. Хуулагч байгуулагч нь обьект руу заалтаар нь хандаж түүний хуулбар обьектыг зөвхөн нэг удаа үүсгэнэ.

Ач холбогдол: Шинээр байгуулж байгаа обьектод дахин гарааны утга оноох үйлдэл хийхгүйгээр үйлдлийг нь хялбарчилна, мөн өмнөх обьектын утга өөрчлөгдөхөд даган өөрчлөгдөнө.

4. Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?

Энэ нь хуулагч байгуулагчтай төстэй боловч хуулагч байгуулагч зөвхөн нэг удаа ашиглагддаг бол хуулагч функцийг хэдэн ч удаа дуудаж ашиглаж болно.

Динамикаар нөөцөлсөн санах ойгоо чөлөөлж байж санах ойн цоорхойгоос сэргийлнэ.

```
Employee *worker =new Employee;
delete worker;
```

5. Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? new операторoop санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

Объектын хаяган хувьсагч зарлахдаа:

```
Employee *ptr;
```

new операторoop санах ой нөөцлөх:

```
ptr = new Employee;
```

Объектын гишүүн өгөгдөл, гишүүн функцэд хандахдаа 2 янзаар хандана:

(хаяган хувьсагчийн нэр).гишүүн өгөгдөл/функц;

(*ptr).name;

хаяган хувьсагчийн нэр . гишүүн өгөгдөл/функц;

ptr.name;

6. Объектын хаяган хувьсагчийн хүснэгт үүсгэх жишээ код бичиж ажиллуул.

```
Employee *emp = new Employee[2];
for(i = 0; i < 2; i++) {
     cout << emp[i].getId() << endl;
     cout << emp[i].getName() << endl;
     cout << emp[i].getPosition() << endl;
     cout << emp[i].getHoursWorked() << endl;
}
```

```
❯ cd "/Users/javhln/Des
0
name
worker
0
0
name
worker
0
```

Хэрэгжүүлэлт:

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

class Employee {
  private:
     int id;
     char *name;
     char *position;
     float hoursWorked;
  public:
     Employee() {
        id = 0;
        name = new char[1];
        strcpy(name, " ");
        position = new char[1];
        strcpy(position, " ");
```

```
        hoursWorked = 0;
    }
    Employee(int idx, char *fname, char *pos, float hrs) {
        id = idx;
        name = new char[strlen(fname) + 1];
        position = new char[strlen(pos) + 1];
        strcpy(name, fname);
        strcpy(position, pos);
        hoursWorked = hrs;
    }
    Employee(Employee &emp) {
        id = emp.id;
        strcpy(name, emp.name);
        strcpy(position, emp.position);
        hoursWorked = emp.hoursWorked;
    }
    int getId() {
        return id;
    }
    char getName() {
        return *name;
    }
    char getPosition() {
        return *position;
    }
    int getHoursWorked() {
        return hoursWorked;
    }
    void setId(int i){
        id = i;
    }
    void setName(char *fname) {
        strcpy(name, fname);
    }
    void setPosition(char *pos) {
        strcpy(position, pos);
    }
    void setHoursWorked(float hrs) {
        hoursWorked = hrs;
    }
```

```
    float calSalary() {
        if(strcmp(position, "zahiral") == 0) {
            return salaryBoss();
        }
        return hoursWorked * 5;
    }
    void inputData() {
        cout << "Id:"; cin >> id;
        cout << "Name:"; cin >> name;
        cout << "Position:";
        cin >> position;
        cout << "Hours:";
        cin >> hoursWorked;
    }
    void showData() {
        cout << "Id:" << id << endl;
        cout << "Name:" << name << endl;
        cout << "Position:" << position << endl;
        cout << "Hours:" << hoursWorked << endl;
    }

  public:
    bool incHoursWork(float hrs) {
        if(hrs ≥ 0 && hrs ≤ 24) {
            hoursWorked += hrs;
            return 1;
        }
        return 0;
    }

    ~Employee() {
        delete name;
        delete position;
    }

  private:
    float salaryBoss() {
        return hoursWorked * 8;
    }
};
```

```
int main()
{
  int i, j;
  Employee *emp = new Employee [3];

  for (i = 0; i < 3; i++)
  {
     cout << "Shine Ajilchinii medeelliig oruulna uu: \n";
     emp[i].inputData();

     while (1)
     {
        int q = 0;
        for (j = 0; j < i; j++)
        {
           if (emp[i].getId() == emp[j].getId())
           {
              q++;
           }
        }
        if (q == 0)
           break;
        else
        {
           cout << "!!!Ajilchinii ID davhtsaj bn. Dahin medeellee
oruulna uu. \n";
           emp[i].inputData();
        }
     }
  }
```

```
  for (i = 0; i < 3; i++)
  {
     for (j = i; j < 3; j++)
     {
        if (emp[i].getName() > emp[j].getName())
        {
           Employee temp;
           temp = emp[i];
           emp[i] = emp[j];
           emp[j] = temp;
        }
     }
  }

  for (i = 0; i < 3; i++)
  {
     cout << "Ajilchid: \n";
     emp[i].showData();
  }

  return 0;
}
```