

C++ Programming	Student number	21300691
Homework 3	Name	Cheung, Won Sik

Header file and Student structure that have name, id, room number, penalty point

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string> // getline
#include <cstdlib> // system pause
#include <sstream> // istringstream
#include <ctype.h> // isalpha
#include <cctype> // toupper
using namespace std; // Or use std::cout

// declare student structure that includes student name, id, room number,
penalty point
typedef struct _student{
    string name;
    string student_id;
    string room;
    int penalty;
}_Student;
```

Dormitory class that have student list, number and methods

```
// declare dormitory class to modify student list
class Dormitory{
private:
    int numStudent;
    Student* student;
public:
    // Initialize dormitory class
    void setInfo(int num, Student* stu);
    // search student list and return index of student
    int searchStudent(string name);
    // get index of student than penalty that index's student
    void penaltyStudent(int index);
    // delete student from student list
    void deleteStudent(int index);
    // free student list that allocated dynamically
    void freeStudent();
    // print student list
    void printStudent();
    // save student list to file
    void saveStudent(string fName);
    // give user interfaces to check student should be penalized
    void checkStudent();
};
```

Functions that are used in main and other functions.

```
// To know student number read line numbers from file
int getLineNum(string fName);
// get student information from file
Student* makeStudent(int numStudent, string fName);
// check is c same as check1, check2 or not
int checkSelection(char c, char check1, char check2);
```

Main function

```
int main(void){
    // declare file name
    string fname = "dormitory.txt";

    // get line number to know student number, and student list
    int line = getLineNum(fname);
    Student* student = makeStudent(line, fname);

    // declare dormitory object, initialize and print student list
    Dormitory dorm;
    dorm.setInfo(line, student);
    dorm.printStudent();

    // give user interface to check student should be penalized
    dorm.checkStudent();

    // finally, print student list, save it to file and free student list
    // that allocated dynamically
    dorm.printStudent();
    dorm.saveStudent(fname);
    dorm.freeStudent();

    system("pause");
    return 0;
}
```

getLineNum function

```
// To know student number read line numbers from file
int getLineNum(string fName){
    ifstream inData;
    inData.open(fName);
    //inData.open("dormitory.txt", ios::in);

    if(!inData){
        cout<<"File opening error"<<endl;
        return -1;
    }

    int line = 0;
    string str = "";
    // getline function get line from stream into string
    while(getline(inData, str)){
        line++;
    }
    inData.close();
    return line-1; // except first line
}
```

makeStudent function

```
// get student information from file
Student* makeStudent(int numStudent, string fName){
    ifstream inData;
    inData.open(fName);
    //inData.open("dormitory.txt", ios::in);

    if(!inData){
        cout<<"File opening error"<<endl;
        return NULL;
    }

    cout<<"open file"<<endl;
    cout<<fName<<endl;

    // make Student structure pointer array dynamically allocated
    Student* student = new Student[numStudent];
    string str = "";

    getline(inData, str); // except first line
    for(int i=0; i<numStudent; i++){
        string index;
        getline(inData, str);
        // make string to token
        istringstream iss(str);
        iss >> student[i].name;
        iss >> student[i].student_id;
        iss >> student[i].room;
        iss >> index;
        student[i].penalty = atoi(index.c_str());
    }

    inData.close();

    return student; // except first line
}
```

checkSelection function

```
// check is c same as check1, check2 or not
int checkSelection(char c, char check1, char check2){
    if(!isalpha(c))
        return -1;

    //make upper alphabet
    c = toupper(c);

    //return true if c = check1
    //return false if c = check2
    //return error if c does not coincide with check1 or check2
    if(c==check1)
        return 1;
    else if(c==check2)
        return 0;
    else
        return -1;
}
```

Dormitory::setInfo function

```
// Initialize dormitory class
void Dormitory::setInfo(int num, Student* stu){
    numStudent = num;
    student = stu;
}
```

Dormitory::penaltyStudent function

Dormitory::deleteStudent function

```
// get index of student than penalty that index's student
void Dormitory::penaltyStudent(int index){
    int penalty = student[index].penalty;

    //if penalty will be 10 point delete student from list
    if(penalty >= 9){
        deleteStudent(index);
    }
    else{
        student[index].penalty++;
    }
}

// delete student from student list
void Dormitory::deleteStudent(int index){
    Student* newstudent = new Student[numStudent-1];

    //feed student information to new student entry
    for(int i=0, j=0; i<numStudent; i++){
        if(i==index)
            continue;
        else{
            newstudent[j] = student[i];
            j++;
        }
    }

    freeStudent();
    student = newstudent;
    numStudent--;
}
```

Dormitory::freeStudent function

Dormitory::printStudent function

```
// free student list that allocated dynamically
void Dormitory::freeStudent(){
    delete[] student;
}

// print student list
void Dormitory::printStudent(){
    cout<<left<<setw(10)<<"name"
        <<left<<setw(10)<<"studentid"
        <<left<<setw(5)<<"room"
        <<left<<setw(12)<<"penaltypoint" <<endl;

    for(int i=0; i<numStudent; i++){
        cout<<left<<setw(10)<<student[i].name
            <<left<<setw(10)<<student[i].student_id
            <<left<<setw(5)<<student[i].room
            <<left<<setw(12)<<student[i].penalty<<endl;
    }
    cout<<endl;
    return;
}
```

Dormitory::saveStudent function

```
// save student list to file
void Dormitory::saveStudent(string fName){
    ofstream outData;
    outData.open(fName);
    //outData.open("test.txt", ios::out);

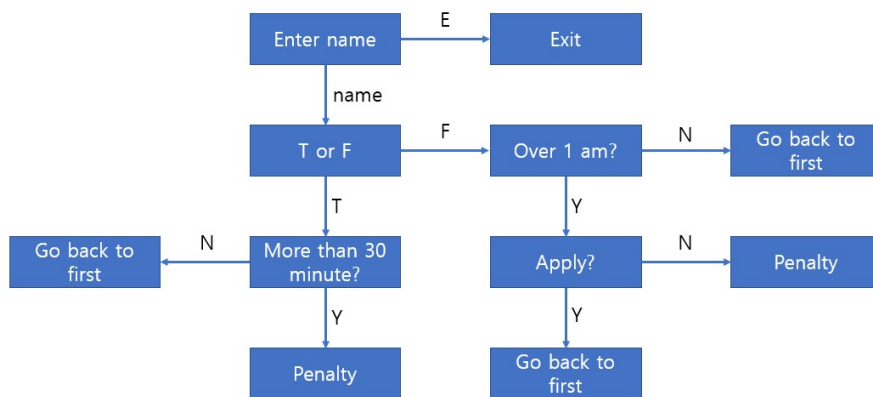
    if(!outData){
        cout<<"File opening error"<<endl;
        return;
    }

    outData<<left<<setw(10)<<"name"
        <<left<<setw(10)<<"studentid"
        <<left<<setw(5)<<"room"
        <<left<<setw(12)<<"penaltypoint" <<endl;

    for(int i=0; i<numStudent; i++){
        outData<<left<<setw(10)<<student[i].name
            <<left<<setw(10)<<student[i].student_id
            <<left<<setw(5)<<student[i].room
            <<left<<setw(12)<<student[i].penalty<<endl;
    }

    outData.close();
}
```


The selection of the user interface is configured as follows.



Dormitory::checkStudent function

```

// give user interfaces to check student should be penalized
void Dormitory::checkStudent(){
    while(1){
        string input;
        bool loop=true;

        // first question
        cout<<"Who tried tagging? Or exit(E/e)"<<endl;
        cin>>input;
        // search student index
        int index = searchStudent(input);
        if(input=="E" || input=="e"){
            break;
        }

        if(index == -1){
            cout<<"Can not find student"<<endl;
            continue;
        }
        else{
            while(loop){
                char opt;
                cout<<"Did "<<input<<" go out after tagging? Or did you return to your dormitory for the first time today? T(t)/F(f)"<<endl;
                cin>>opt;
                // check user's input
                int check1 = checkSelection(opt, 'T', 'F');
                if(check1==-1)
                    continue;
                else if(check1){

```


Result

```
open file
dormitory.txt
name      studentid room penaltypoint
Hwang     21400123  313    9
Choi      21500456  206    2
Kim       21200145  416    2
Shim      21500321  116    2
Lee       21600999  308    2
Jung      21300777  407    3

Who tried tagging? Or exit(E/e)
Hwang
Did Hwang go out after tagging? Or did you return to your dormitory for the first time today? T(t)/F(f)
T
Did it take more than 30 minutes? Y(y)/N(n)
Y
Hwang was penalized for 1 point
```

```
Who tried tagging? Or exit(E/e)
Hwang
Can not find student
```

```
Who tried tagging? Or exit(E/e)
Choi
Did Choi go out after tagging? Or did you return to your dormitory for the first time today? T(t)/F(f)
T
Did it take more than 30 minutes? Y(y)/N(n)
N
```

```
Who tried tagging? Or exit(E/e)
Kim
Did Kim go out after tagging? Or did you return to your dormitory for the first time today? T(t)/F(f)
F
Is it over 1 am right now? Y(y)/N(n)
Y
Did you apply for out-of-home application? Y(y)/N(n)
N
Kim was penalized for 1 point
Who tried tagging? Or exit(E/e)
Shim
Did Shim go out after tagging? Or did you return to your dormitory for the first time today? T(t)/F(f)
F
Is it over 1 am right now? Y(y)/N(n)
Y
Did you apply for out-of-home application? Y(y)/N(n)
Y
Who tried tagging? Or exit(E/e)
Jung
Did Jung go out after tagging? Or did you return to your dormitory for the first time today? T(t)/F(f)
F
Is it over 1 am right now? Y(y)/N(n)
N
```

```
Who tried tagging? Or exit(E/e)
e
name      studentid room penaltypoint
Choi      21500456  206    2
Kim       21200145  416    3
Shim      21500321  116    2
Lee       21600999  308    2
Jung      21300777  407    3

계속하려면 아무 키나 누르십시오 . . .
```