

Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018

LinkedList ve HashTable ile Dosya Yönetimi:

LinkedList:

```
1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 using namespace std;
5
6 struct Student {
7     string name;
8     int age;
9     string major;
10
11     Student(string n, int a, string m) {
12         this->name = n;
13         this->age = a;
14         this->major = m;
15     }
16 };
17
18 class LinkedList {
19 private:
20     struct Node {
21         Student student;
22         Node* next;
23
24         Node(Student s) : student(s) {
25             this->student = s;
26             this->next = nullptr;
27         }
28     };
29
30     Node* head;
31
32 public:
33     LinkedList(){
34         this->head = nullptr;
35     }
36
37     void insert(Student student) {
38         Node* newNode = new Node(student);
```

```
37
38     void insert(Student student) {
39         Node* newNode = new Node(student);
40
41         if (head == nullptr) {
42             head = newNode;
43         }
44         else {
45             Node* curr = head;
46             while (curr->next != nullptr) {
47                 curr = curr->next;
48             }
49             curr->next = newNode;
50         }
51     }
52
53     void update(string name, int newAge, string newMajor) {
54         Node* curr = head;
55         while (curr != nullptr) {
56             if (curr->student.name == name) {
57                 curr->student.age = newAge;
58                 curr->student.major = newMajor;
59                 break;
60             }
61             curr = curr->next;
62         }
63     }
64
65     void remove(string name) {
66         if (head == nullptr)
67             return;
68
69         if (head->student.name == name) {
70             Node* temp = head;
71             head = head->next;
72             delete temp;
73             return;
74         }
```

Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018

```
FileOrganizationHomework3.1

main.cpp x  oğrencibilgileri.txt +
main.cpp > f printToFile
Format Run
Console x Shell AI +
Ask AI 3s on 02:06:09, 04/28 ✓

75 Node* curr = head;
76 while (curr->next != nullptr) {
77     if (curr->next->student.name == name) {
78         Node* temp = curr->next;
79         curr->next = curr->next->next;
80         delete temp;
81         return;
82     }
83     curr = curr->next;
84 }
85
86 void printToFile(string filename) {
87     ofstream file(filename);
88     if (file.is_open()) {
89         Node* curr = head;
90         while (curr != nullptr) {
91             file << curr->student.name << " " << curr->student.age << " " << curr->student.major << endl;
92             curr = curr->next;
93         }
94         file.close();
95     } else {
96         cout << "Dosya açilamadı" << endl;
97     }
98 }
99
100 };
101 LinkedList dosyadanOku(string filename){
102     LinkedList list;
103
104     ifstream file("ogrencibilgileri.txt");
105     if (file.is_open()) {
106         string line;
107         while (getline(file, line)) {
108             stringstream Line(line);
109             string name, major;
110             int age;
111             getline(Line, name, ',');
112             getline(Line, major, ',');
113             age = stoi(Line.str());
114             Student student(name, age, major);
115             list.insert(student);
116         }
117         file.close();
118     } else {
119         cout << "Dosya açilamadı" << endl;
120     }
121 }
122 return list;
123 };
124
125 int main() {
126     LinkedList linkedList = dosyadanOku("ogrencibilgileri.txt");
127
128     Student newStudent("Mustafa Çirci", 20, "Computer Science");
129     linkedList.insert(newStudent);
130
131     linkedList.update("Ahmet Yılmaz", 21, "Mathematics");
132
133     linkedList.remove("Ahmet Yılmaz");
134
135     linkedList.printToFile("ogrencibilgileri.txt");
136
137     return 0;
138 }
139
140 }
```

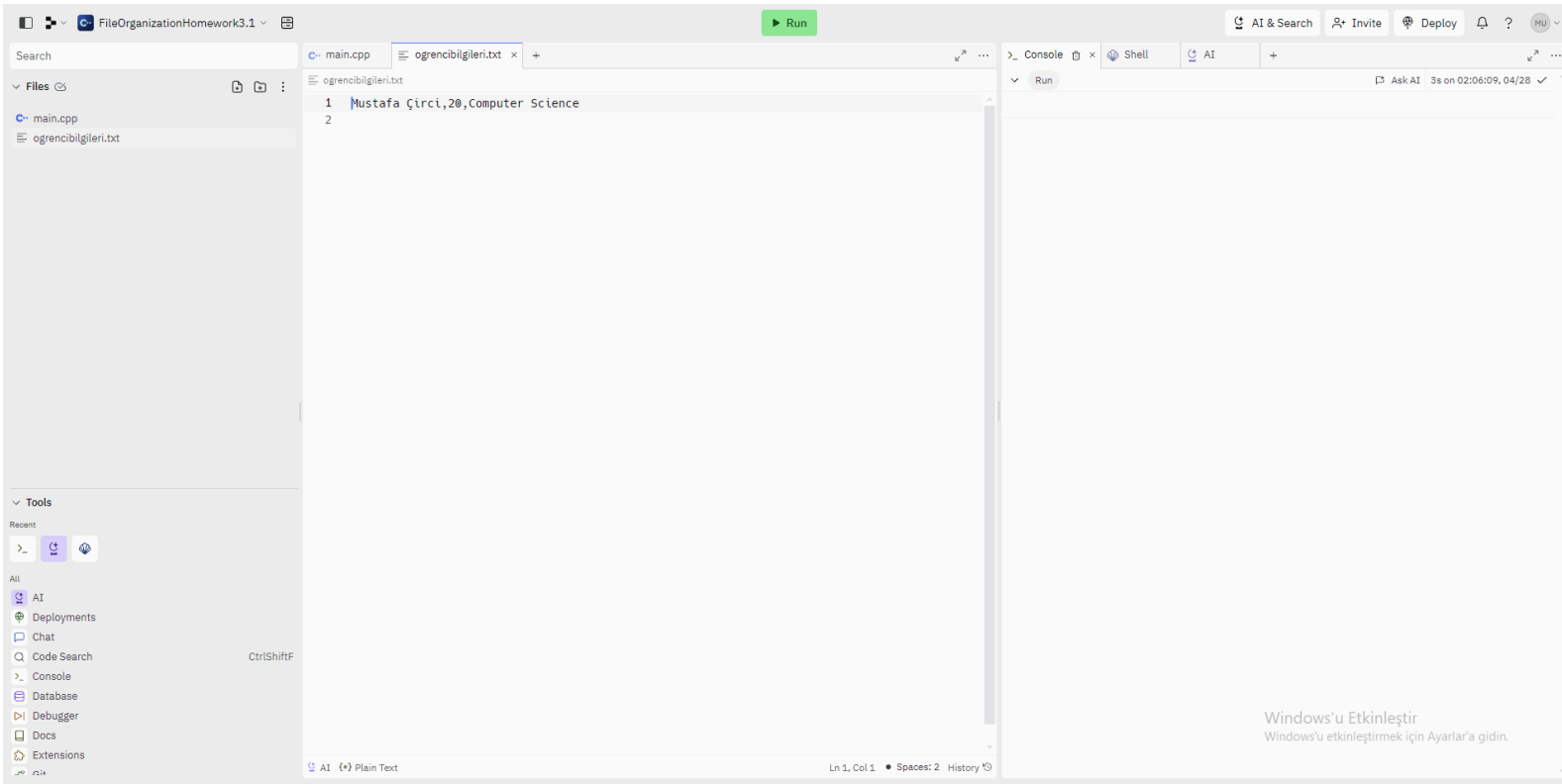
```
FileOrganizationHomework3.1

main.cpp x  oğrencibilgileri.txt +
main.cpp > f printToFile
Format Run
Console x Shell AI +
Ask AI 3s on 02:06:09, 04/28 ✓

103
104 ifstream file("ogrencibilgileri.txt");
105 if (file.is_open()) {
106     string line;
107     while (getline(file, line)) {
108         stringstream Line(line);
109         string name, major;
110         int age;
111         getline(Line, name, ',');
112         Line >> age;
113         Line.ignore();
114         getline(Line, major, ',');
115         Student student(name, age, major);
116         list.insert(student);
117     }
118     file.close();
119 }
120 else {
121     cout << "Dosya açilamadı" << endl;
122 }
123 return list;
124 };
125
126 int main() {
127     LinkedList linkedList = dosyadanOku("ogrencibilgileri.txt");
128
129     Student newStudent("Mustafa Çirci", 20, "Computer Science");
130     linkedList.insert(newStudent);
131
132     linkedList.update("Ahmet Yılmaz", 21, "Mathematics");
133
134     linkedList.remove("Ahmet Yılmaz");
135
136     linkedList.printToFile("ogrencibilgileri.txt");
137
138     return 0;
139 }
140 }
```

Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018



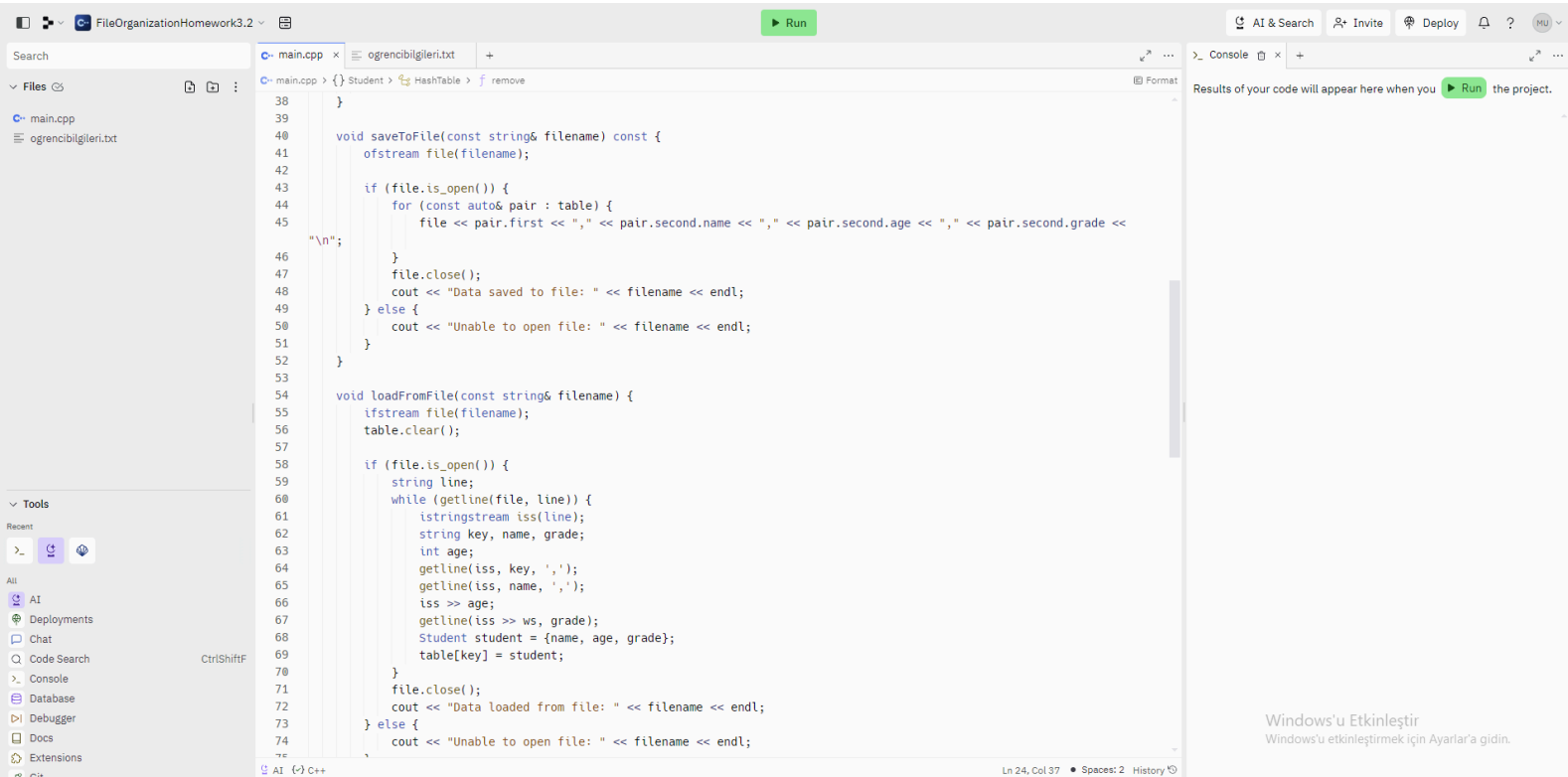
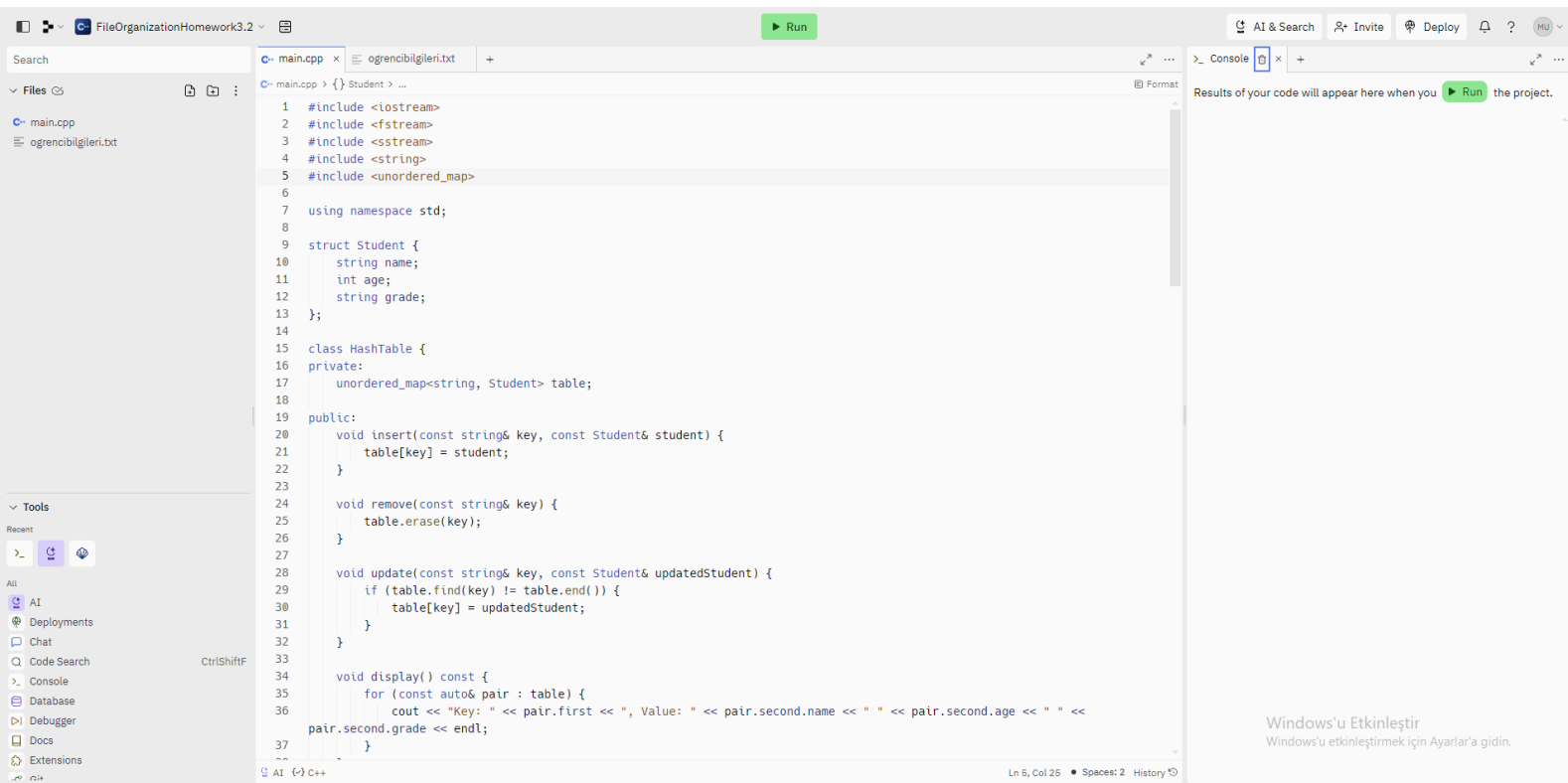
```
1 Mustafa Çirci,20,Computer Science
2
```

1. Öncelikle öğrencileri bir arada tutmak için Student adında bir struct oluşturdum.
2. Daha sonrasında LinkedList sınıfının içerisinde her bir düğümü tutmak için Node adında struct oluşturdum.
3. LinkedList sınıfı constructor oluşturduktan sonra LinkedList de gerçekleştireceğim her bir için ayrı ayrı fonksiyon oluşturdum.
4. Insert fonksiyonu öncelikle aktif olan Node objesini kontrol eder. Eğer obje boş ise yeni oluşturulan objeyi oraya kayıt eder. Eğer obje doluysa teker teker Node objesinin next parametresini dolaşır. Bulunan ilk boş yere oluşturulan obje kayıt edilir.
5. Update fonksiyonu fonksiyona verilen parametre değerindeki ilgili obje bulunana kadar bütün objeleri döngü ile dolaşır. Objeyi bulduktan gerekli güncellemeleri yapar.
6. Remove fonksiyonu fonksiyona parametresi verilen objeyi bulmak için döngü halinde bütün düğümleri dolaşır. İlgili düğüm bulunduktan sonra önceki düğümün next değeri, ilgili objenin next değerine eşitlenir. Böylece düğüm aradan çıkarılmış ve silinmiş olur.
7. Son olarak printToFile fonksiyonu ile LinkedList de bulunan bütün düğümler dolaşılır ve dosyaya yazılır.
8. dosyadanOku fonksiyonu ile ilgili dosya açılır, ve her satırı LinkedList'e düğüm olarak kayıt edilir.

Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018

HashTable:



Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018

The screenshot shows a C++ IDE with a project named "FileOrganizationHomework3.2". The main.cpp file contains the following code:

```
1 // HashTable implementation
2 #include <iostream>
3 #include <string>
4 #include <vector>
5 #include <map>
6 #include <fstream>
7 using namespace std;
8
9 struct Student {
10     string name;
11     int age;
12     string grade;
13 };
14
15 class HashTable {
16 public:
17     map<string, Student> table;
18     void loadFromFile(string filename) {
19         fstream file(filename, ios::in);
20         if (file.is_open()) {
21             string line;
22             while (getline(file, line)) {
23                 iss >> age;
24                 getline(iss >> ws, grade);
25                 Student student = {name, age, grade};
26                 table[key] = student;
27             }
28             file.close();
29             cout << "Data loaded from file: " << filename << endl;
30         } else {
31             cout << "Unable to open file: " << filename << endl;
32         }
33     }
34 };
35
36 int main() {
37     HashTable hashtable;
38     hashtable.loadFromFile("ogrencibilgileri.txt");
39
40     Student student1 = {"Mustafa Çirci", 20, "A"};
41     hashtable.insert("1306210018", student1);
42
43     Student updatedStudent = {"Ahmet Yılmaz", 21, "B"};
44     hashtable.update("1306240001", updatedStudent);
45
46     hashtable.remove("1306210018");
47
48     Student student2 = {"Mustafa Çirci", 20, "A"};
49     hashtable.insert("1306210018", student2);
50
51     hashtable.display();
52
53     hashtable.saveToFile("ogrencibilgileri.txt");
54
55     return 0;
56 }
```

The IDE interface includes a search bar, a file explorer on the left showing "main.cpp" and "ogrencibilgileri.txt", and a tools panel with options like AI, Deployments, Chat, Code Search, Console, Database, Debugger, Docs, and Extensions. The status bar at the bottom indicates "Ln 61, Col 18" and "Spaces: 2".

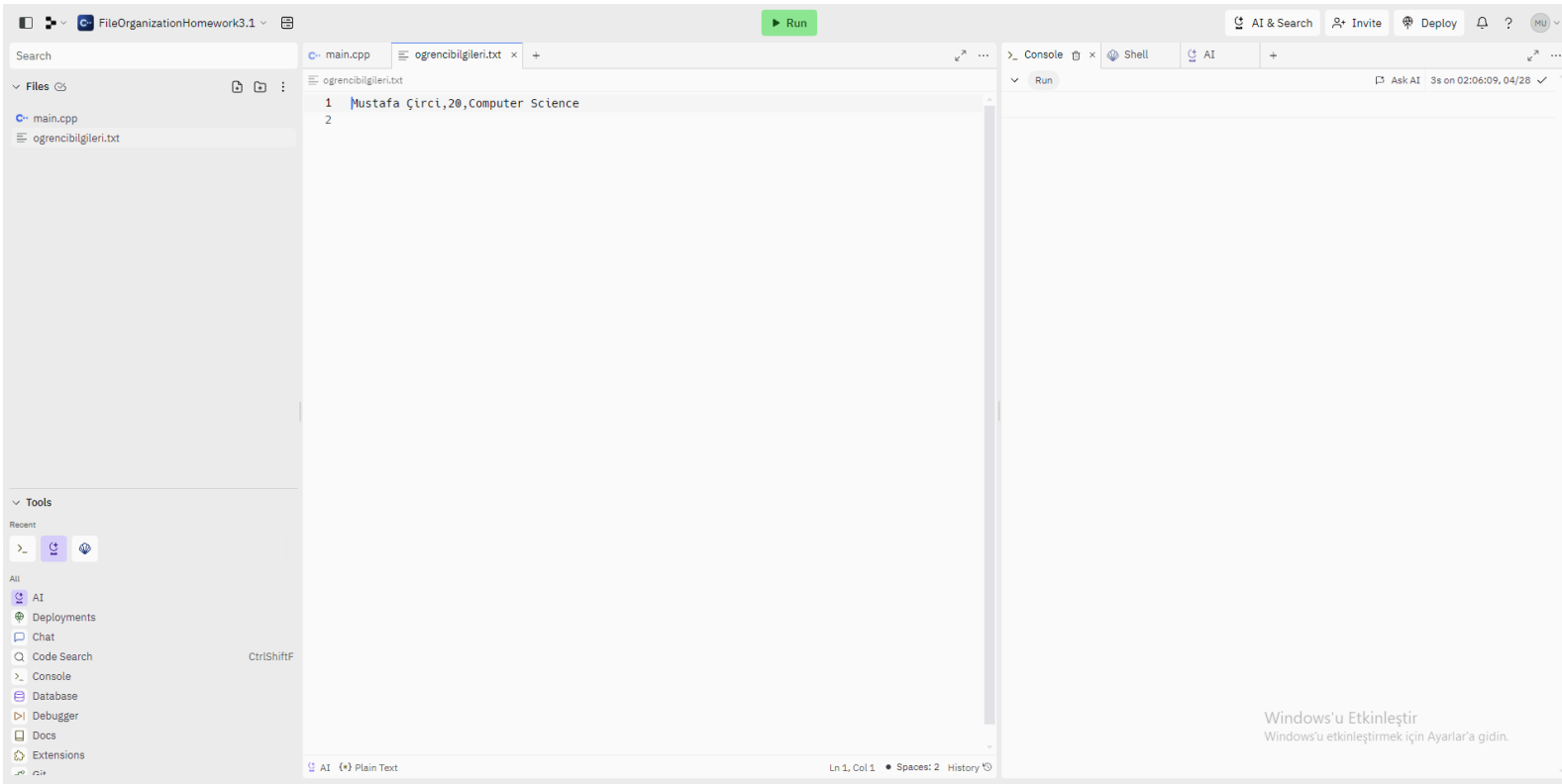
The screenshot shows the same C++ IDE after running the program. The console output displays the following data:

```
1 1306210018,Mustafa Çirci,20,A
2
```

The IDE interface is similar to the previous screenshot, but the status bar at the bottom now indicates "Ln 1, Col 30" and "Spaces: 2". The right sidebar shows a message: "Results of your code will appear here when you Run the project."

Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018



```
1 Mustafa Çirci,20,Computer Science
2
```

1. Öncelikle öğrenci bilgilerini bir arada tutmak için Student adında bir struct oluşturdum.
2. Daha sonrasında HashTable adında sınıf oluşturdum. Bu sınıf içerisinde unordered_map kullanarak table adında bir map oluşturdum.
3. Insert fonksiyonu ile oluşturduğum map objesine verilen key elementini kullanarak student objesi ekledim.
4. Remove fonksiyonu ile verilmiş olan key elementine ait bilgiyi erase komutu ile mapten sildim.
5. Update fonksiyonu ile verilmiş olan key değerine ait objeyi find fonksiyonu ile buldum. Daha sonrasında da ilgili objeye ait verileri güncelledim.
6. Display fonksiyonu ile de her bir elemanın yazdırılmasını sağladım.
7. saveToFile fonksiyonu ile mapteki her bir elemanı gezip ilgili değerlerin dosyaya yazdırılmasını sağladım.
8. loadFromFile fonksiyonu ile verilmiş olan dosya adına ait dosyadan, her bir satırı map'e ekler.