```
/*
Name : Mahesh Gaikwad
Roll No : SA21
Assignment 12 : Direct Access File
Implementation of a direct access file -Insertion and deletion of a record
from a direct access file
*/
#include <iostream>
#include <fstream> #include <cstring> using namespace std;
const int MAX_RECORDS = 100; // Maximum number of records
const int HASH_SIZE = 10; // Hash table size
const int MAX_NAME_LENGTH = 20; // Maximum length for name const
int MAX_ADDRESS_LENGTH = 50; // Maximum length for address struct
Record
{
int id;
char name[MAX_NAME_LENGTH]; char
address[MAX_ADDRESS_LENGTH];
Record() : id(0)
{
memset(name, 0, sizeof(name)); memset(address,
0, sizeof(address));
} };
class HashFile
{
private: string
filename;
fstream file;
int hashTable[HASH_SIZE];
public:
```

```
HashFile(const string &filename) : filename(filename) {}

void initialize(); void insertRecord(const

Record &record); void deleteRecord(int

id); void displayRecords(); int

hashFunction(int id);

};

void HashFile::initialize()

{

// Clear the hash table for (int i

= 0; i < HASH_SIZE; ++i)

 {

hashTable[i] = -1;

}

// Create and open the file in binary mode

file.open(filename, ios::binary | ios::in | ios::out | ios::trunc);

if (!file)

{

cout << "Failed to open the file." << endl; return;

}

// Initialize the file with empty records Record

emptyRecord;

for (int i = 0; i < MAX_RECORDS; ++i)

{

file.write(reinterpret_cast<const char *>(&emptyRecord),
sizeof(Record));

} file.close(); cout << "Hash file initialized

successfully." << endl;

}

void HashFile::insertRecord(const Record &record)

{
```

```
// Calculate the hash value using the record ID int
hashValue = hashFunction(record.id); // Open
the file in binary mode file.open(filename,
ios::binary | ios::in | ios::out);
if (!file)
{
cout << "Failed to open the file." << endl; return;
}
// Move the file pointer to the appropriate location based on the
hash value file.seekp(hashValue * sizeof(Record), ios::beg); //
Write the record at the current file position
file.write(reinterpret_cast<const char *>(&record),
sizeof(Record)); // Update the hash table entry
hashTable[hashValue] = record.id;
file.close(); cout << "Record inserted
successfully." << endl;
}
void HashFile::deleteRecord(int id)
{

// Calculate the hash value using the record ID
int hashValue = hashFunction(id); // Open the file
in binary mode file.open(filename, ios::binary |
ios::in | ios::out);
if (!file)
{
cout << "Failed to open the file." << endl; return;
}
// Move the file pointer to the appropriate location based on the
hash value file.seekp(hashValue * sizeof(Record), ios::beg);
```

```
// Create an empty record to overwrite the deleted record
Record emptyRecord;
// Write the empty record at the current file position
file.write(reinterpret_cast<const char *>(&emptyRecord),
sizeof(Record)); // Update the hash table entry
hashTable[hashValue] = -1;
file.close();
cout << "Record deleted successfully." << endl;
}
void HashFile::displayRecords()
{
// Open the file in binary mode for reading file.open(filename,
ios::binary | ios::in);
if (!file)
{
cout << "Failed to open the file." << endl; return;
}
// Read and display each record from the file
Record record;
for (int i = 0; i < HASH_SIZE; ++i)
{
if (hashTable[i] != -1)
{
// Move the file pointer to the appropriate location

based on the hash value

file.seekg(i * sizeof(Record), ios::beg); //
Read the record at the current file position
file.read(reinterpret_cast<char *>(&record),
sizeof(Record));

// Display the record cout << "Hash
Value: " << i << endl;

cout << "ID: " << record.id << endl; cout <<
"Name: " << record.name << endl; cout <<
"Address: " << record.address << endl; cout <<
endl;
}
}
file.close();
}
int HashFile::hashFunction(int id)
{
return id % HASH_SIZE;
}
int main()
{
HashFile hashFile("Assignment12.bin");
hashFile.initialize(); int choice = 0;
while (choice != 4)
{ cout << "-------------------------------------" <<
endl;
cout << "Direct Access File Menu:" << endl;
cout << "1. Insert Record" << endl; cout <<
"2. Delete Record" << endl; cout << "3.
Display Records" << endl; cout << "4. Quit"
<< endl; cout << "Enter your choice: "; cin
>> choice; switch (choice)
{
```

```
case 1:
{
Record record; cout << "Enter
ID: "; cin >> record.id; cout <<
"Enter Name: "; cin >>
record.name; cout << "Enter
Address: "; cin >>
record.address;
hashFile.insertRecord(record);
break;
}
case 2:
{ int
id;
cout
<<
"Ente
r the
ID of
the
recor
d to
delet
e: ";

cin >> id;
hashFile.deleteRecord(id); break;
}
case 3:
{
```

```
hashFile.displayRecords(); break;
}
case 4:
{
cout << "Quitting..." << endl; break;
}
default: cout << "Invalid choice. Please try again."
<< endl; break;
}
}
return 0;
}
```

OUTPUT

```
-------------------------------------
Direct Access File Menu:
1. Insert Record
2. Delete Record
3. Display Records 4. Quit
Enter your choice: 1
Enter ID: 19
Enter Name: Teju
Enter Address: Warje
Record inserted successfully.
-------------------------------------
Direct Access File Menu:
1. Insert Record
2. Delete Record
3. Display Records
4. Quit

Enter your choice: 1
Enter ID: 06

Enter Name: Sneha

Enter Address: Room

Record inserted successfully.

-------------------------------------

Direct Access File Menu:

1. Insert Record

2. Delete Record

3. Display Records 4. Quit

Enter your choice: 1

Enter ID: 35

Enter Name: Max

Enter Address: Hostel

Record inserted successfully.

-------------------------------------

Direct Access File Menu:

1. Insert Record

2. Delete Record

3. Display Records

4. Quit

Enter your choice: 1

Enter ID: 23

Enter Name: Raj

Enter Address: Pune

Record inserted successfully.

-------------------------------------

Direct Access File Menu:

1. Insert Record

2. Delete Record

3. Display Records

4. Quit
```

```
Enter your choice: 3

Hash Value: 3

ID: 23

Name: Raj

Address: Pune

Hash Value: 5

ID: 35

Name: Max

Address: Hostel

Hash Value: 6


ID: 6

Name: Sneha

Address: Room

Hash Value: 9

ID: 19

Name: Teju

Address: Warje -------------------------------------

Direct Access File Menu:

1. Insert Record

2. Delete Record

3. Display Records

4. Quit

Enter your choice: 2 Enter the ID of

the record to delete: 23

Record deleted successfully.

-------------------------------------

Direct Access File Menu:

1. Insert Record

2. Delete Record

3. Display Records 4. Quit
```

```
Enter your choice: 3

Hash Value: 5

ID: 35

Name: Max

Address: Hostel

Hash Value: 6

ID: 6

Name: Sneha

Address: Room

Hash Value: 9

ID: 19

Name: Teju

Address: Warje

-------------------------------------

Direct Access File Menu:

1. Insert Record

2. Delete Record

3. Display Records
```