

Assignment - 10

Title : File Handling

Problem : Department maintains student information. The file contains roll no., name, div and address. Allows user to add, delete, search student data. If record does not exist print particular message. Use sequential file for implementing file handling.

Objective : To study and implement file handling operations.

Outcome : Studied and implemented file handling operations using sequential file.

Theory : File is a stream of bytes. Size of file is expressed in number of bytes.
Sequential File : In this file, records are added in order of arrival. Length of record is not fixed. Search is time consuming. Insertion and Deletion is time consuming.

Random File : We can read/write a particular word without having read permission records, we can position file pointer to a particular location.

Modes :

ios :: in Read from file.

ios :: out Write to file.

ios :: ate go to end.

ios :: app append to end of file.

ios :: trunc truncate file.

ios :: binary treat file as binary.

ALGORITHM

1. Write to file.

```
Write (int roll, char name, int div)
```

```
{
```

```
Student s = new W(roll, name, class)
```

```
file.seekp(0, ios::end)
```

```
file.write((char*)&s, sizeof(s));
```

```
}
```

2. display

```
display ()
```

```
{
```

```
Student s;
```

```
file.seekg(0, ios::beg);
```

```
while (file.read((char*)&s, sizeof(s)));
```

```
{ s.display data(); }
```

```
}
```


3.) delete

```
void delete(int roll)
```

```
{
```

```
    ofstream newfile("new.txt", ios::bin)
```

```
    file.seekg(0, ios::beg);
```

```
    bool flag = False;
```

```
    student s;
```

```
    while (not end of file)
```

```
    {
```

```
        if (s.roll == roll) then
```

```
        {
```

```
            flag = true;
```

```
            continue;
```

```
        }
```

```
    newfile.write((char*)&s, size of(s))
```

```
    }
```

```
    if (flag)
```

```
    {
```

```
        print("Record not found");
```

```
        file.close();
```

```
        remove("Student.txt");
```

```
        remove
```

```
        rename("new.txt", "student.txt");
```

```
        file.open("Student.txt", ios::bin);
```

```
    }
```

4. Search

```

void search(int roll)
{
    student s;
    file = seekg(0, ios::beg);
    while (file has not ended) do
    {
        if (s.roll == roll) then
        {
            s.display data();
            break;
        }
    }
}

```

TEST CASES

I/P	Expected O/P	ACTUAL O/P	RESULT
1. Roll no : 1, 22, 33 Name : abc, aa, bc Div : 2, 3, 1	1 abc 2 22 aa 3 33 bc 1	1 abc 2 22 aa 3 33 bc 1	Success
2. Search : 22	22 aa 3	22 aa 3	Success
3. Delete : 22	1 abc 2 33 bc 1	1 abc 2 33 bc 1	Success

Conclusion: We have successfully implemented file handling operations using sequential file.