

SPPU-SE-COMP-CONTENT – KSKA Git

Modern Education Society's Wadia College of Engineering, Pune

210256: DATA STRUCTURES and ALGORITHM LABORATORY (2019 COURSE)

NAME OF STUDENT: Tanmay Machken	CLASS: Comp I
SEMESTER/YEAR: 9E 4th Sem	ROLL NO: F22111017
DATE OF PERFORMANCE:	DATE OF SUBMISSION:
EXAMINED BY:	EXPERIMENT NO: 11

TITLE: Create sequential file and maintain data as specified

AIM/PROBLEM STATEMENT: Department maintains a student information. The file contains roll number, name, division, and address. Allow user to add delete information of student. Display information of particular student. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to maintain the data.

OBJECTIVES:

1. To understand file handling.
2. To understand working of sequential file.

OUTCOMES:

1. To apply appropriate file handling techniques on given data
2. To use sequential file.

PRE-REQUISITE:

1. Knowledge of C++ programming
2. Basic knowledge of sequential file

THEORY:

File is a collection of records related to each other. The file size is limited by the size of memory and storage medium.

There are two important features of file:

1. **File Activity:** File activity specifies percent of actual records which proceed in a single run.

SPPU-SE-COMP-CONTENT - KSKA Git

* Sequential File access organization:

- Storing and sorting in contiguous block within files on tape or disk is called as sequential access.
- In sequential access file organization, all records are stored in sequential order. The records are arranged in ascending or descending order of a key field.
- Sequential File search starts from the beginning of the file and the records can be added at the end of the file.

→ This method can be implemented in two ways:-

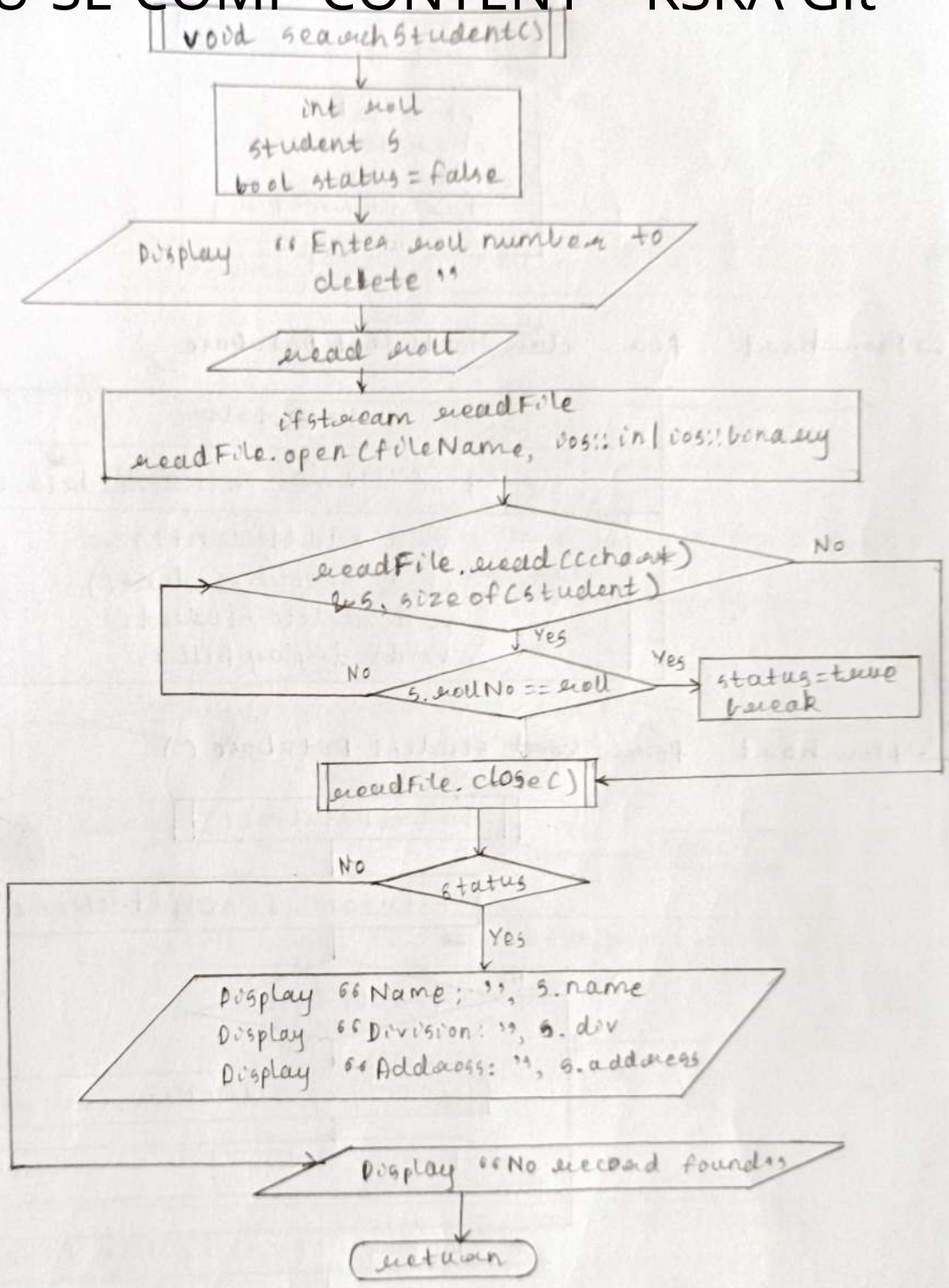
1. Plain File method:

- In this method, we store the record in sequence, i.e., one after another.
- Hence, the record will be inserted in the order in which they are inserted into tables.

2. Sorted file method:

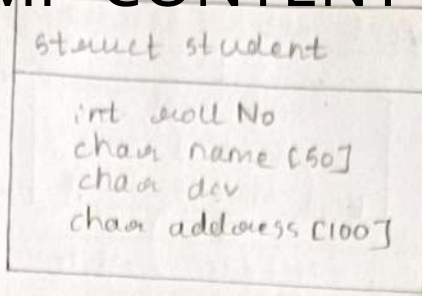
- In this method, the new record is always inserted at the file's end, and then it will sort the sequence in ascending or descending order.
- Sorting of records is based on any primary key or any other key.

void searchStudent() SPPU-SE-COMP-CONTENT - KSKA Git

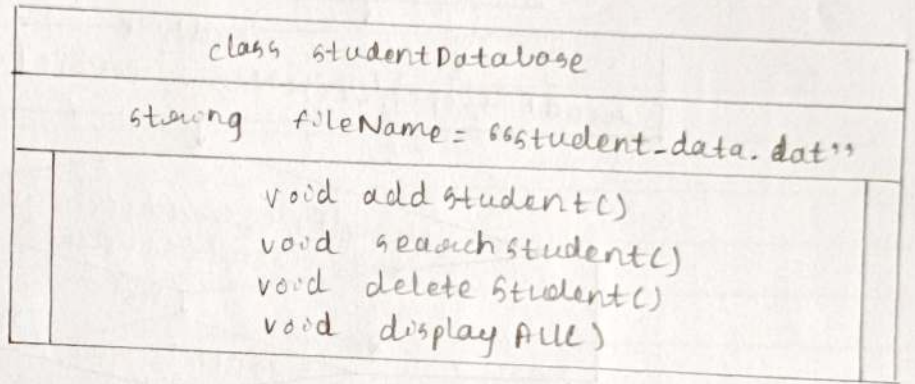


→ Flowchart for student student

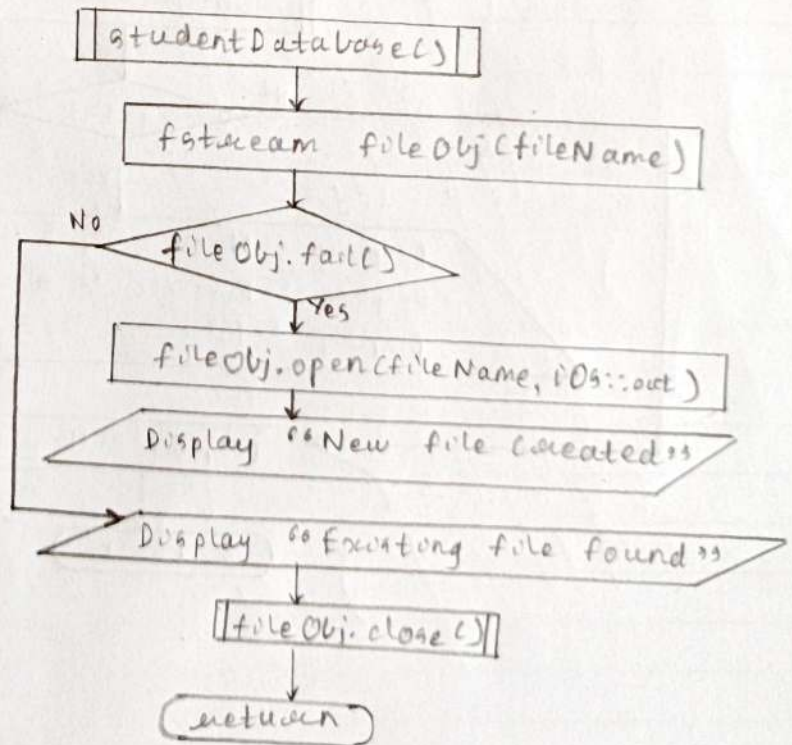
SPPU-SE-COMP-CONTENT - KSKA Git



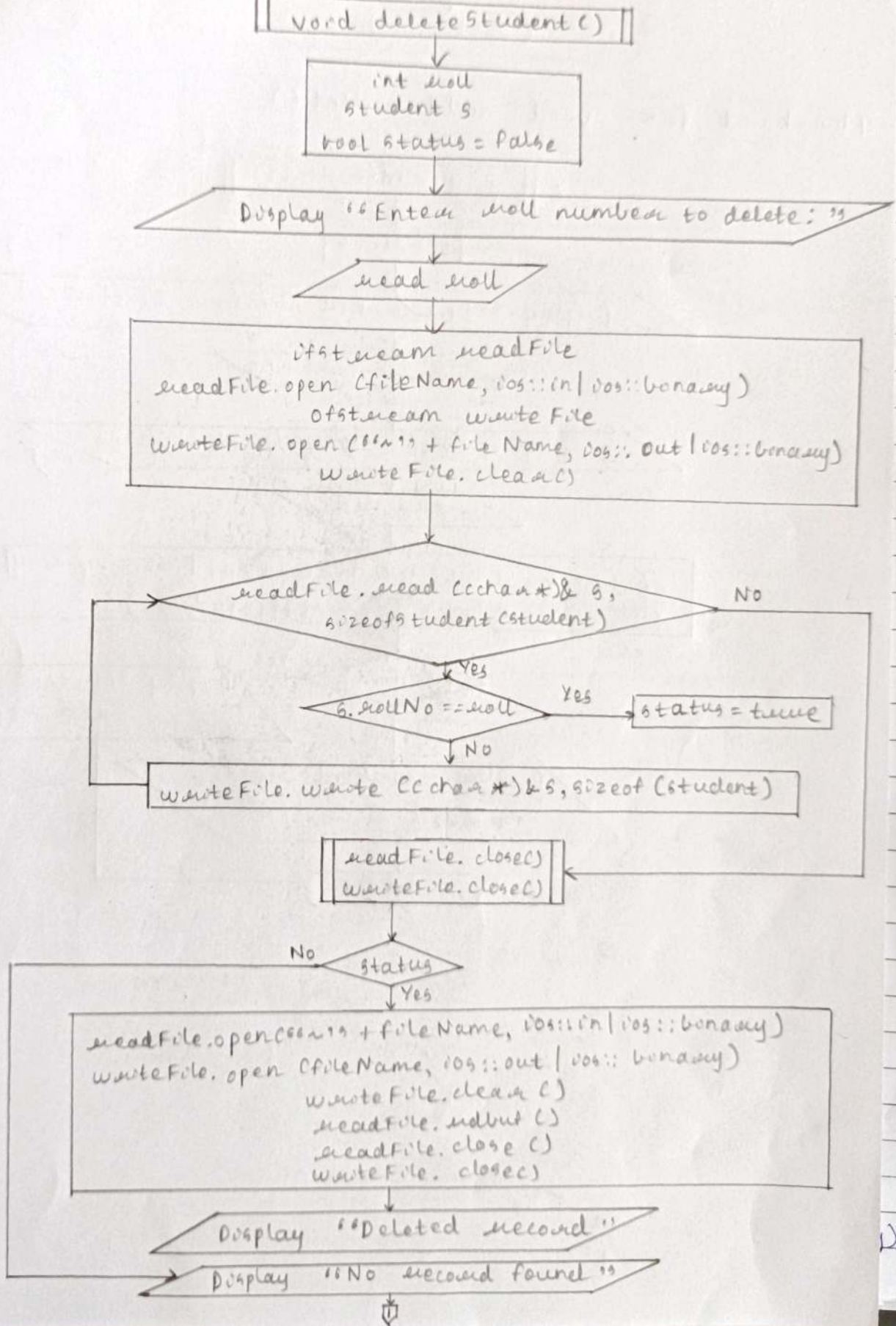
→ Flowchart for class student Database



→ Flowchart for ~~void~~ student Database ()



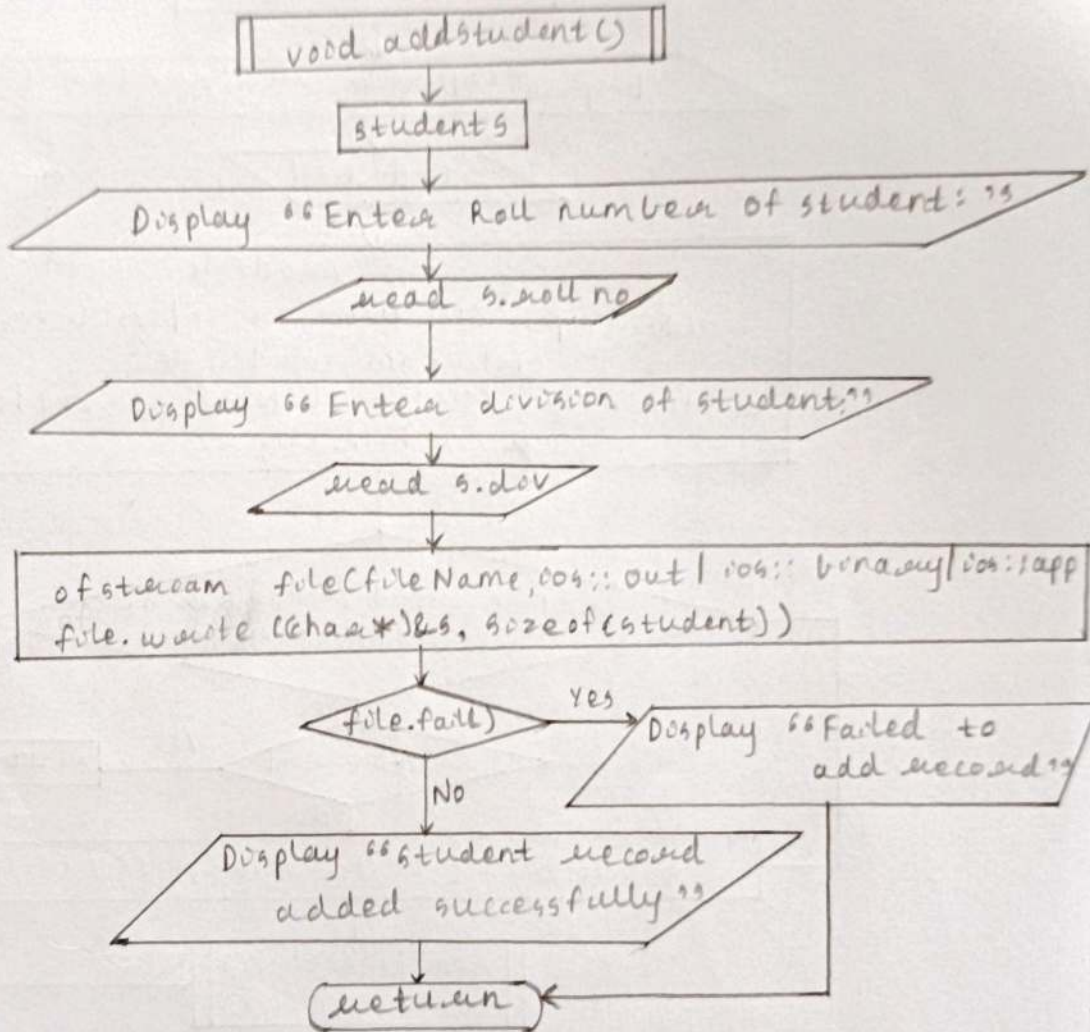
SPPU-SE-COMP-CONTENT - KSKA Git



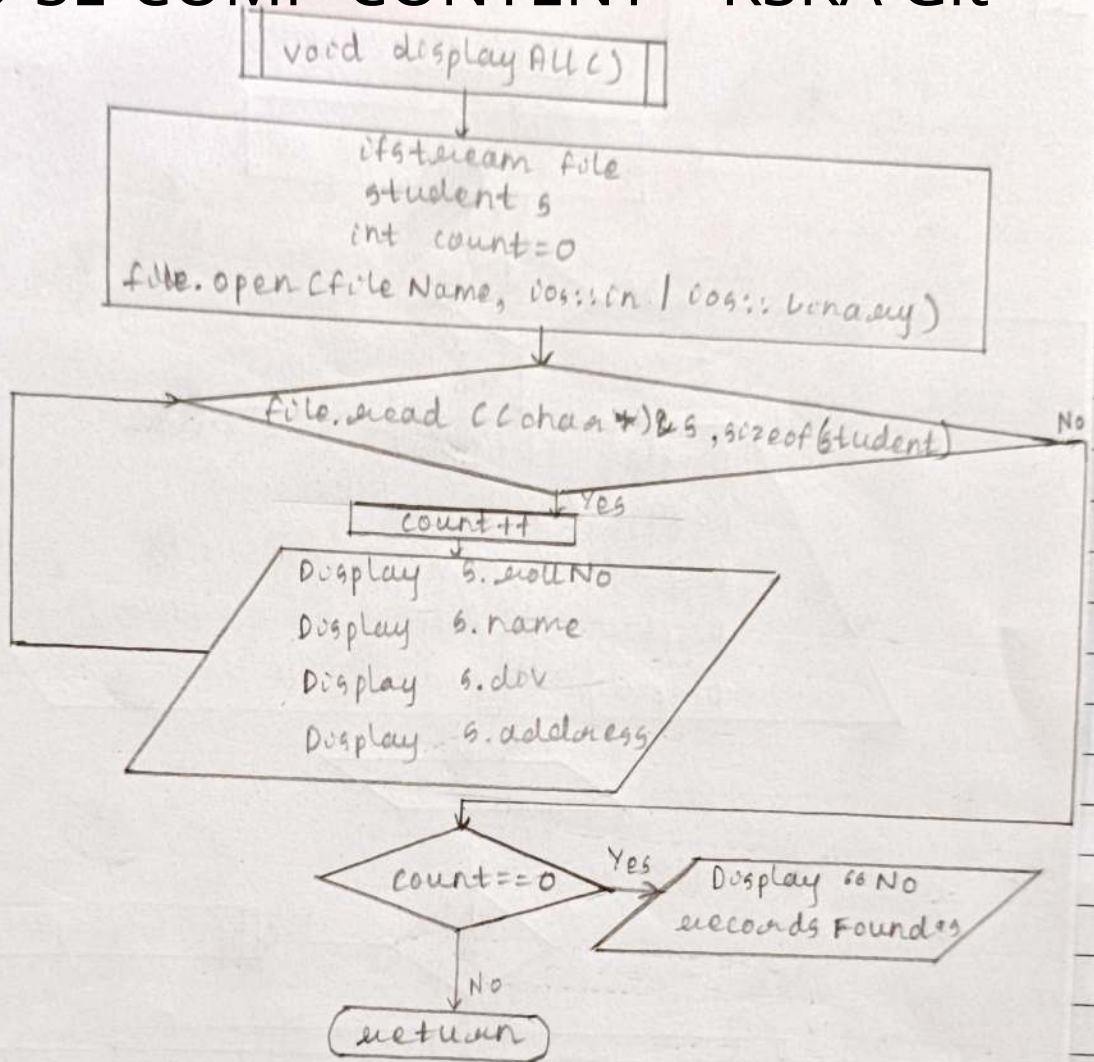
SPPU-SE-COMP-CONTENT - KSKA Git

return

→ Flowchart for void addstudent()

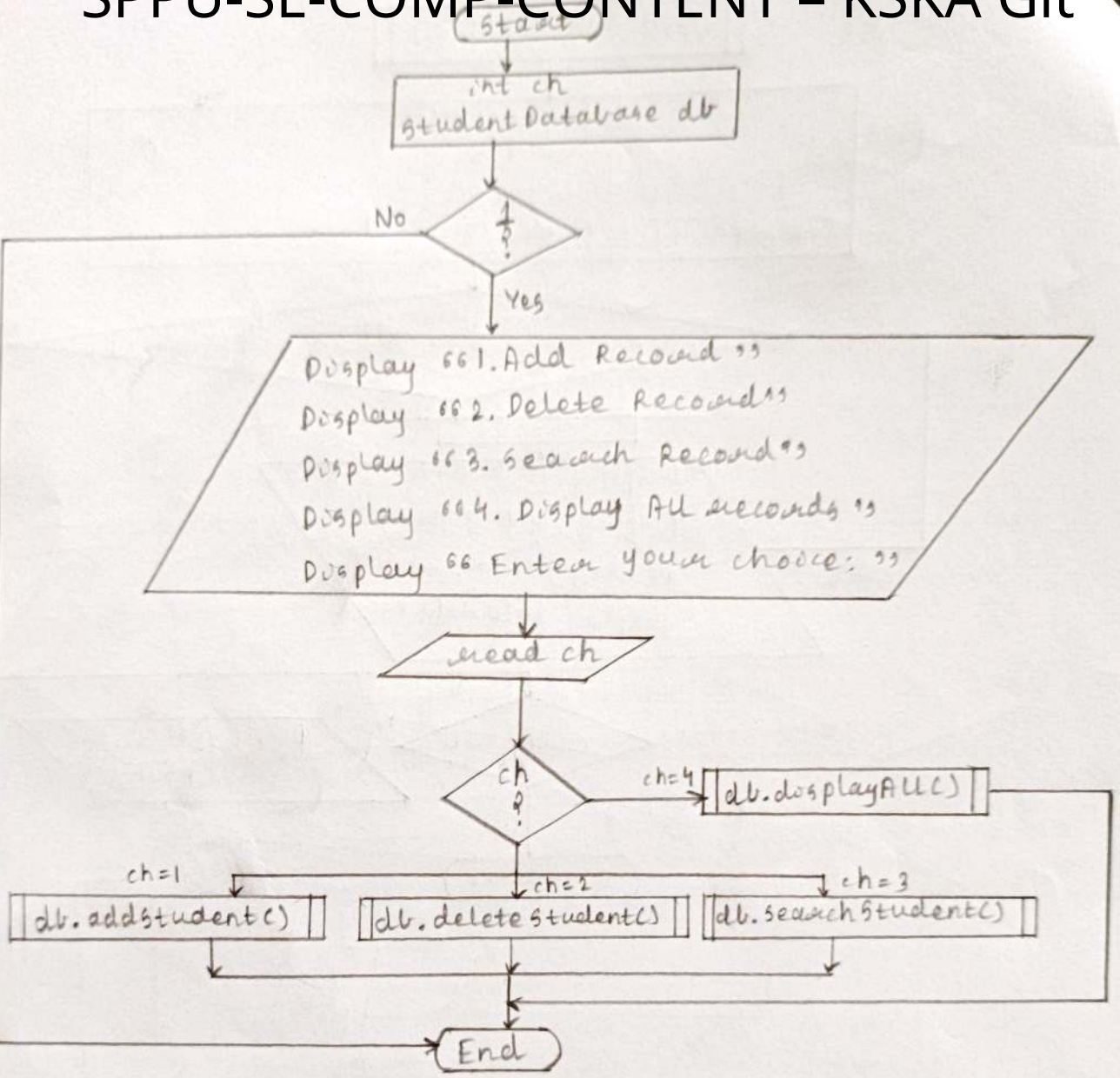


SPPU-SE-COMP-CONTENT - KSKA Git



→ Flowchart for int main()

SPPU-SE-COMP-CONTENT – KSKA Git



SPPU-SE-COMP-CONTENT - KSKA Git

→ Pseudocode for struct student

1. Declare `int rollNo`
`char name[50]`
`char div`
`char address[100]`
2. create student

→ Pseudocode for class student Database

1. Declare `string fileName = "student_data.dat"`
2. create `student Database()`

begin

`ifstream fileObj(fileName)`

`if fileObj.fail() then`

`fileObj.open(fileName, ios::out)`

`Display "New file Created"`

`else`

`Display "Existing File Found"`

`call function fileObj.close()`

end

3. Declare `void addstudent()`
`void searchstudent()`
`void deletestudent()`
`void displayAll()`

→ Pseudocode for `void searchStudent()`

1. Declare `int roll`
`student s`
2. Initialize `bool status = false`
3. Display `"Enter roll number to delete:"`
4. read roll
5. create `ifstream readfile`
6. Declare `readfile.open(fileName, ios::in | ios::binary)`

SPPU-SE-COMP-CONTENT - KSKA Git

```
7. while fread File.read (char* & s, size of (student)) do
begin
    if s.roll No == roll then
        initialize status status = true
        break
    end
end
8. call function readfile.close()
9. if status then
    Display "Found record with details"
    Display "Roll No:", s.roll No
    Display "Name:", s.name
    Display "Division:", s.div
    Display "Address:", s.address
else
    Display "No record found"
10. return

→ Pseudocode for void delete student()
1. Declare int roll
    student s
2. initialize bool status = false
3. Display "Enter roll number to delete:"
4. read roll
5. create ifstream readfile
6. create readfile.open (fileName, ios::in | ios::binary)
7. create ofstream writefile
8. create writefile.open ("~" + fileName, ios::out |
    ios::binary)
9. writefile.close()
10. while readfile.read (char* & s, sizeof (student)) do
begin
    if s.roll No == roll then
```

SPPU-SE-COMP-CONTENT - KSKA Git

```
        initialize status = true
    else
        create writeFile, write ( (char*) &s,
                                sizeof (student), flush
    end
11. call readFile.close()
    writeFile.close()
12. if status then
    create readFile.open ("r" + file Name, ios::in |
                            ios::binary)
    create writeFile.open (file Name, ios::out |
                            ios::binary)
    call writeFile.clear()
    readFile.close()
    writeFile.close()
    Display "Deleted record"
else
    Display "No record found"
13. return
```

→ Pseudocode addStudent ()

1. int student s
2. Display "Enter roll number of student: "
3. read s. rollNo
4. Display "Enter name of student: "
5. cin.ignore ()
cin.getline (&s, address, 100)
6. Create ofstream file (file Name, ios::out | ios::binary |
ios::app)
7. call file.write ((char*) &s, sizeof (student))
8. if file.fail () then
Display "Failed to add record"

SPPU-SE-COMP-CONTENT - KSKA Git

else

Display "Student record added successfully"

9. call file.close()

10. return

→

→ Pseudocode for void display All()

1. create ifstream file

2. create student s

3. initialize int count = 0

4. call file.open(fileName, ios::in | ios::binary)

5. while (file.read((char*) &s, sizeof student)) do
begin
~~int~~ increment count

Display count

Display s.rollNo

Display s.name

Display s.div

Display s.address

end

6. if count == 0 then

Display "No records Found"

7. call file.close()

8. return

→ Pseudocode for int main()

1. case declare int ch

2. create student Database db

3. while (1) do

begin

Display "1. Add Record:"

Display "2. Delete Record:"

Display "3. Search Record:"

Display "4. Display All records"

SPPU-SE-COMP-CONTENT - KSKA Git

```
Display "Enter your choice"  
read ch  
switch (ch) then
```

```
case 1:  
  create create db.addStudent()  
  break
```

```
case 2:  
  create delete student() db.addSt.  
  break
```

```
case 3:  
  create db.searchStudent()  
  break
```

```
case 4:  
  Display "Records on File are"  
  create db.displayAll()  
  break
```

```
default:
```

```
  Display "Enter a valid choice"  
  break
```

```
end
```

```
4. End
```

SPPU-SE-COMP-CONTENT - KSKA Git

Q1. Explain direct sequential file.

Ans. It is the most common type of file. In this type of file:

- i) A fixed format is used for record.
- ii) All records are of the same length.
- iii) Position of each field in record and length of field is fixed.
- iv) Records are physically ordered on the value of ~~field~~ one of the fields, called the ordering fields.

Block 1

Name	Roll no.	Year	Marks
Tanmay, Anut	1000	1	60.48
Tanmay, Pratat	1010	2	53.98
Agawal, Mohan	1012	1	70.43
Vishwa, Tanmay	1015	3	65.4

Block 2

Tanmay	1016	4	68.90
--------	------	---	-------

→ Advantages:-

- Reading of records in order of the ordering key is extremely efficient.
- Finding the next record in order of the ordering key usually, does not require additional block access.

→ Disadvantages:-

- Sequential file ~~are~~ does not give any advantage when the search operation is to be carried out on non-ordering field.
- Deleting a ~~record~~ ^{record} is an expensive operation.

Q2. Explain advantage and disadvantage of ordering ~~index~~.

~~Advantages~~ sequential file.

Ans. 1. Advantages:- Sequential file can be searched effectively on

SPPU-SE-COMP-CONTENT - KSKA Git

ordering key.

- when it is necessary to search for a record on the basis of some other attribute than the ordering key field, sequential file representation is inadequate
 - multiple indices can be maintained for each type of field to be used for searching. Thus, indexing provides much better flexibility.
 - 1. An index file ~~is~~ usually requires less storage space than the main file.
 - A binary search on sequential file will require accessing of more blocks.
- Disadvantage:-
- Requires additional storage space for the index which can increase the cost and complexity of the system.
 - Implementing and maintaining the index for sequential file organization can be complex, especially as the dataset grows.

Q3. Explain advantages and disadvantages of direct access method.

Ans. → Advantages:-

1. Speed:

- Direct access method provides fast access to the data, as it enables direct access to a specific location on memory.

2. Simplicity:

- The method is simple as it does not require any intermediate processing.

3. Flexibility:

- It offers flexibility as the memory can be

SPPU-SE-COMP-CONTENT - KSKA Git

accessed on any order, not just sequentially.

→ Disadvantages:-

1. Complexity:

- It requires a complex hardware architecture to support a direct access to the memory.

2. Cost:

- The hardware required for direct access is usually more expensive than other memory management methods.

3. Limited memory size:

- The method is limited by the size of the memory that can be directly accessed.

→ Conclusion:-

1. Successfully understood file handling
2. Successfully implemented working of sequential file.