

# FUTURE VISION BIE

One Stop for All Study Materials  
& Lab Programs



*Future Vision*

By K B Hemanth Raj

Scan the QR Code to Visit the Web Page



Or

Visit : <https://hemanthrajhemu.github.io>

Gain Access to All Study Materials according to VTU,  
CSE – Computer Science Engineering,  
ISE – Information Science Engineering,  
ECE - Electronics and Communication Engineering  
& MORE...

Join Telegram to get Instant Updates: [https://bit.ly/VTU\\_TELEGRAM](https://bit.ly/VTU_TELEGRAM)

Contact: MAIL: [futurevisionbie@gmail.com](mailto:futurevisionbie@gmail.com)

INSTAGRAM: [www.instagram.com/hemanthraj\\_hemu/](http://www.instagram.com/hemanthraj_hemu/)

INSTAGRAM: [www.instagram.com/futurevisionbie/](http://www.instagram.com/futurevisionbie/)

WHATSAPP SHARE: <https://bit.ly/FVBIESHARE>

2. Write a C++ program to read and write and student objects with fixed-length records and the fields delimited by "|".implement pack(),unpack(),modify() and search() methods.

### Fixed length record

A record which is predetermined to be the same length as the other records in the file.

Record 1	Record 2	Record 3	Record 4	Record 5
----------	----------	----------	----------	----------

- The file is divided into records of equal size.
- All records within a file have the same size.
- Different files can have different length records.
- Programs which access the file must know the record length.
- Offset, or position, of the nth record of a file can be calculated.
- There is no external overhead for record separation.
- There may be internal fragmentation (unused space within records.)
- There will be no external fragmentation (unused space outside of records) except for deleted records.
- Individual records can always be updated in place

### Delimited Variable Length Fields

Record 1		Record 2		Record 3		Record 4		Record 5
----------	--	----------	--	----------	--	----------	--	----------

- The fields within a record are followed by a delimiting byte or series of bytes.
- Fields within a record can have different sizes.
- Different records can have different length fields.
- Programs which access the record must know the delimiter.
- The delimiter cannot occur within the data.
- If used with delimited records, the field delimiter must be different from the record delimiter.
- There is external overhead for field separation equal to the size of the delimiter per field.
- There should be no internal fragmentation (unused space within fields.)

#### Pack():

This method is used to group all the related field values of particular record taken by the application in buffer.

#### Unpack():

This method is used to ungroup all the related field values of percular record taken from the file in buffer.

**File\_structure2.cpp**

```
#include<iostream.h>
#include<fstream.h>
#include<process.h>
#include<string.h>
#include<conio.h>
class student
{
    private:
        char
        buf[45],name[10],sem[10],branch[10]; public:
        void read()
        {
            cout<<"Name: "<<endl;
            cin>>name;
            cout<<"Semester: "<<endl;
            cin>>sem;
            cout<<"Branch: "<<endl;
            cin>>branch;
        }

        void pack(fstream &ofile)
        {
            read();
            strcpy(buf,"");
            strcat(buf,name);
            strcat(buf,"|");
            strcat(buf,sem);
            strcat(buf,"|");
            strcat(buf,branch);
            strcat(buf,"|");
            while(strlen(buf)<45)
                strcat(buf,"!");
            strcat(buf,"\n");
            ofile.write(buf,strlen(buf));
        }

        void unpack(fstream &ifile)
        {
            char extra[45];
            while(!ifile.eof())
            {
                ifile.getline(name,10,'|');
                ifile.getline(sem,10,'|');
                ifile.getline(branch,10,'|');
```

```
        ifile.getline(extra,45,'\n');
        cout<<name<<"\t"<<sem<<"\t"<<branch<<"\n";
    }
}

int search(fstream &ifile,char key[])
{
    char extra[45];
    while(!ifile.eof())
    {
        ifile.getline(name,10,'|');
        ifile.getline(sem,10,'|');
        ifile.getline(branch,10,'|');
        ifile.getline(extra,45,'\n');
        if(strcmp(name,key)==0)
        {
            cout<<"Record found and details
are:"<<endl; cout<<"Name: "<<name<<endl;
            cout<<"Semester: "<<sem<<endl;
            cout<<"Branch: "<<branch<<endl; return 1;
        }
    }
    return 0;
}

void modify(fstream &iofile,char key[])
{
    if(search(iofile,key))
    {
        cout<<"Record found,enter modification details:"<<endl;
        iofile.seekp(-47,ios::cur);
        pack(iofile);
    }
    else
        cout<<"Sorry!No such record\n";
}

};

void main()
{
    int n,i,ch;
    student stu;
    fstream ofile;
    ofile.open("student.txt",ios::trunc|ios::app);
```

```
    ofile.close();
    clrscr();
    for(;;)
    {
        clrscr();
        cout<<"1. Insert\n2.    Display all\n3.    Search\n4.    Modify\n5.
Exit\n";
        cout<<"Enter your choice"<<endl;
        cin>>ch;
        switch(ch)
        {
            case 1: fstream ofile;
                    ofile.open("student.txt",ios::out|ios::app);
                    cout<<"Enter the no. of students"<<endl;
                    cin>>n;
                    for(i=0;i<n;i++)
                    {
                        stu.pack(ofile);
                    }
                    ofile.close();
                    break;

            case 2: fstream infile;
                    infile.open("student.txt",ios::in)
                    ; stu.unpack(infile);
                    getch();
                    infile.close();
                    break;

            case 3: cout<<"Enter the record name to be
                    searched"<<endl; char key[10];
                    cin>>key;
                    fstream ifile;
                    ifile.open("student.txt",ios::in);
                    if(stu.search(ifile,key)==0)
                        cout<<"record not found\n";
                    getch();
                    ifile.close();
                    break;

            case 4:    fstream iofile;
                    iofile.open("student.txt",ios::in|ios::out);
                    cout<<"Enter the record name to be modified"<<endl;
                    cin>>key;
                    stu.modify(iofile,key);
```

```
        getch();
        iofile.close();
        break;
    default: exit(0);
}
}
```

**Output :**

**1:write to file 2:display the file 3:modify the file 4:search 5.exit**

Enter the choice:1

Enter the number of students:2

Enter the student name = ajay

Enter the sem = 6

Enter the branch = ise

Enter the student name = rahul

Enter the sem = 6

Enter the branch = cse

**1:write to file 2:display the file 3:modify the file 4:search 5.exit**

Enter the choice:2

Name	Sem	Branch
ajay	6	ise
rahul	6	cse

**1:write to file 2:display the file 3:modify the file 4:search**

**5.exit** Enter the choice:4

Enter the record name you want to search =

rahul Record found

rahul 6 cse|

**1:write to file 2:display the file 3:modify the file 4:search**

**5.exit** Enter the choice:3

Enter the record name you want to

modify:rahul record found and details are:

rahul 6 cse|

enter modification details

Enter the student name =navya

Enter the sem = 6

Enter the branch = ise

**1:write to file 2:display the file 3:modify the file 4:search**

**5.exit** Enter the choice:2

Name	Sem	Branch
ajay	6	ise
Navya	6	ise

**1:write to file 2:display the file 3:modify the file 4:search 5.exit**

Enter the choice:4

Enter the record name you want to search:keerthi

Record not found