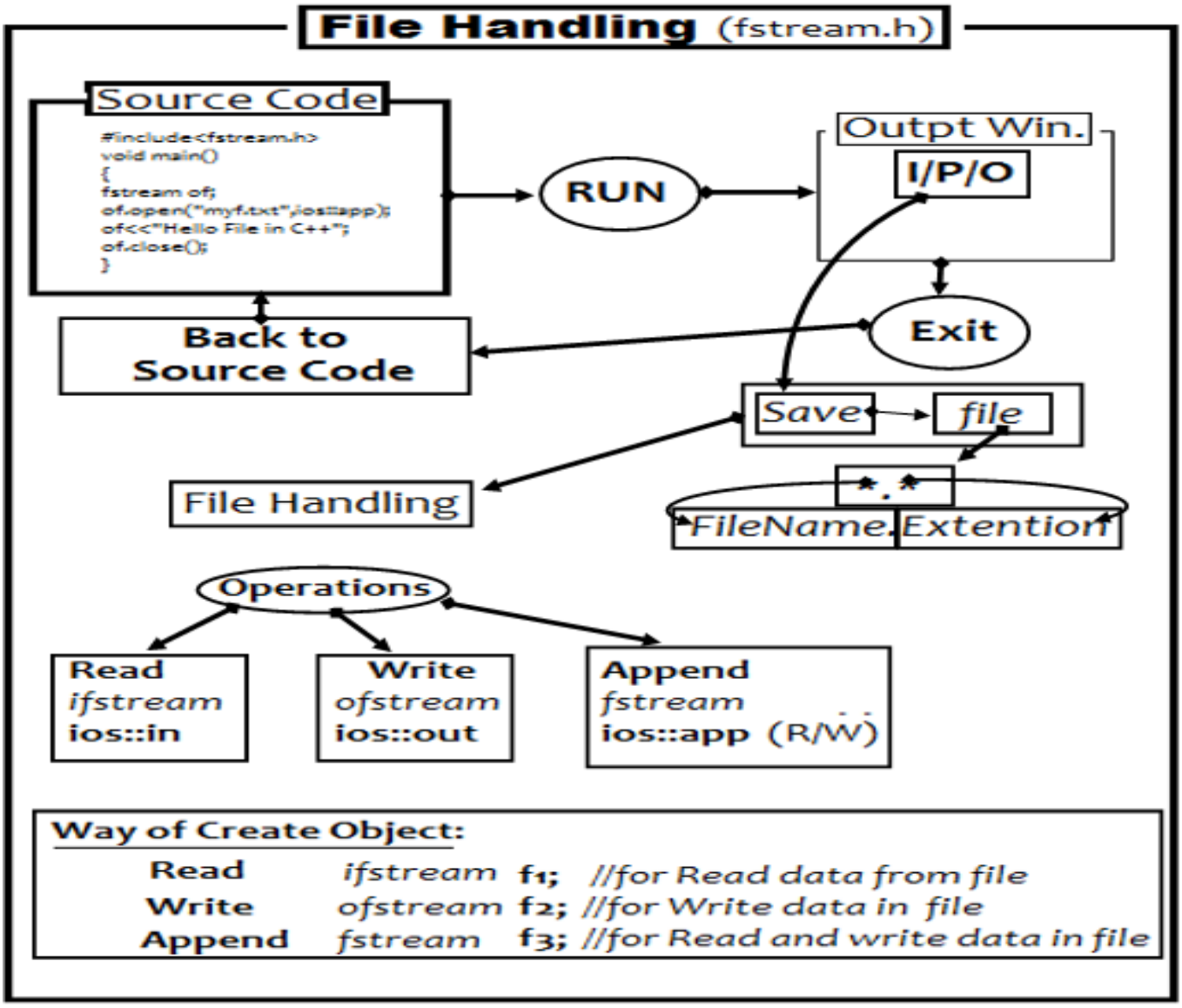


Way of Create Object:

```
Read      ifstream  f1;  //for Read data from file
Write     ofstream  f2;  //for Write data in file
Append    fstream   f3;  //for Read and write data in file
```

Steps to work with files

Step 1: Create object for R/W/App like:

```
Read      ifstream  f1;  //for Read data from file
Write     ofstream  f2;  //for Write data in file
Append    fstream   f3;  //for Read and write data in file
```

Step 2: Open file for R/W/App operation like :

object.open("File Name with Extention ",MODE);

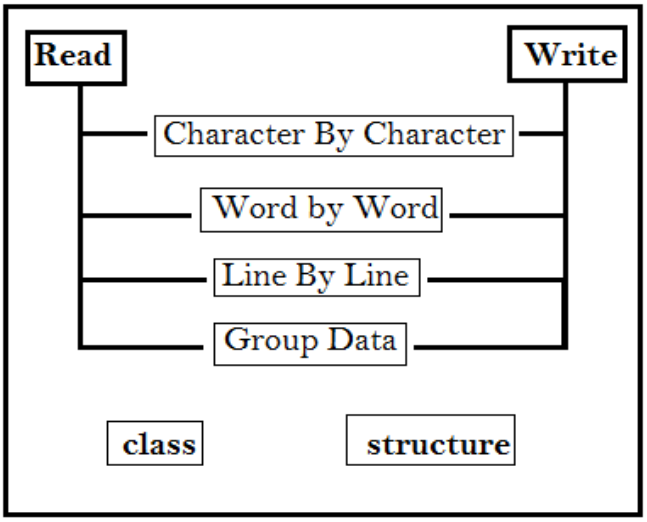
.
file Name.Extention

	ios::in
	ios::out
	ios::app
	ios:: in | ios::binary
If file is a binary file:	ios:: out | ios::binary
	ios:: app | ios::binary

Setp 3: Perform R/W/App Operation help of Variables:

Setp 3: Close existing file by help of objects like:

ObjectName.close();



Write In Data File

```
#include<iostream.h>
#include<fstream.h>
void main()
{
ofstream f1;
f1.open("Mytext.txt",ios::out);
f1<< "This is my C++ Program for handling text files.";
f1.close();
}
```

ofstream f1("Mytext.txt",ios::out);

```
int Age; char Name[30];
cin>>Age; gets(Name);
f1<< Name<<" is my friend. He is "<<Age<< " old";
```

```
f1<< Name<<" is my friend. \n He is "<<Age<< " old";
```

Read

```
#include<iostream.h>
#include<fstream.h>
void main()
{
ifstream f2("Mytext.txt",ios::in);
I: char ch;
II: while(!f2)
{
f2.get(ch);
cout<<ch;
}
f1.close();
}
```

Character By Character

```
c++;
cout<<c;
```

f=1; III:

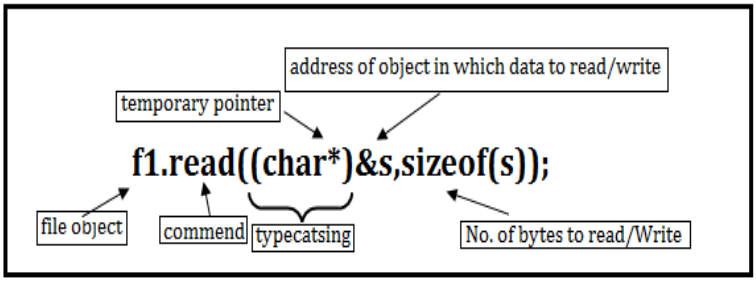
Word by Word

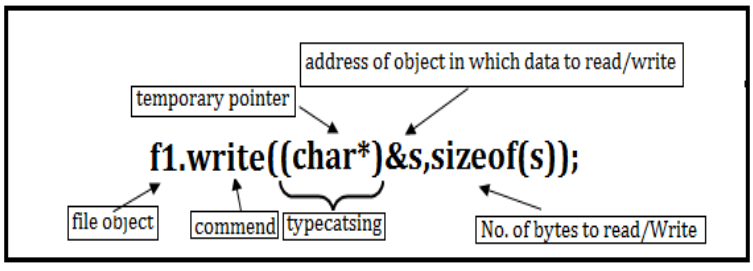
```
I: W[20];
II: while(!f2.eof())
III: f2.getline(W,20,' ');
     f2>>W;
```

LineByLine

```
I: L[80];
II: while(!f2.eof())
III: f2.getline(L,80,'\n');
```

```
if(f==1) cout<<"Yes Present";
else cout<<"Not Present";
```





Example : ***Syntax: write(memory,size);***

For Structure: *fin.write((char*)&[structure Object],sizeof([Structure Name]);*

For Class : *fin.write((char*)&[class Object],sizeof([class Object]);*

Fileobject.write((char *)&object, sizeof(object));

Syntax: read(memory,size);

Example :

For Structure: *fin.read((char*)&[structure Object],sizeof([Structure Name]);*

For Class : *fin.read((char*)&[class Object],sizeof([class Object]);*

Fileobject.read((char *)&object, sizeof(object));

structure

Write Single Record in file

```
void main()
{
DATA D;
ofstream fout;
fout.open("MyData.dat");
cout<<"\n Enter Roll Number :";
cin>>D.RollNo;
cout<<"\n Enter Age :";
cin>>D.Age;
cout<<"\n Enter Name :";
cin>>.DName;
cout<<"\n Enter address :";
cin>>D.Fee;
fout.write((char *)&D,sizeof(DATA));
fout.close();
getch();
}
```

class

```
D.getData();
```

structure

Read Single Record

```
void main()
{
DATA D;
ifstream fin;
fin.open("MyData.dat");
fin.read((char *)&D,sizeof(DATA));
cout<<"\n Roll Number :"<<D.RollNo;
cout<<"\n Name :"<<D.Name;
cout<<"\n Age :"<<D.Age;
cout<<"\n Fee :"<<D.Fee;
fin.close();
getch();
}
```

class

```
D.putData();
```

Read Operations on Group Data

on class and structure

1. *Display Record from file (All/selected)*
2. *Count Record*
3. *Search Record*
4. *Edit Record/Upate/Modify*
5. *Remove Record from file*
6. *Copy record from one file to another*

Display All Record from file

structure

Read Multiple Record

```
void main()
{
DATA D;
ifstream fin;
fin.open("MyData.dat");
While( fin.read((char *)&D,sizeof(DATA)))
   {
f=1;
    cout<<"\n Roll Number :"<<D.RollNo;
    cout<<"\n Name :"<<D.Name;
    cout<<"\n Age :"<<D.Age;
    cout<<"\n Fee :"<<D.Fee;
    c++;
    cout<<c;
   }
fin.close();
getch();
}
```

class

```
D.putData();
```

```
if(f==1) cout<<"Yes Present";
else  cout<<"Not Present";
```

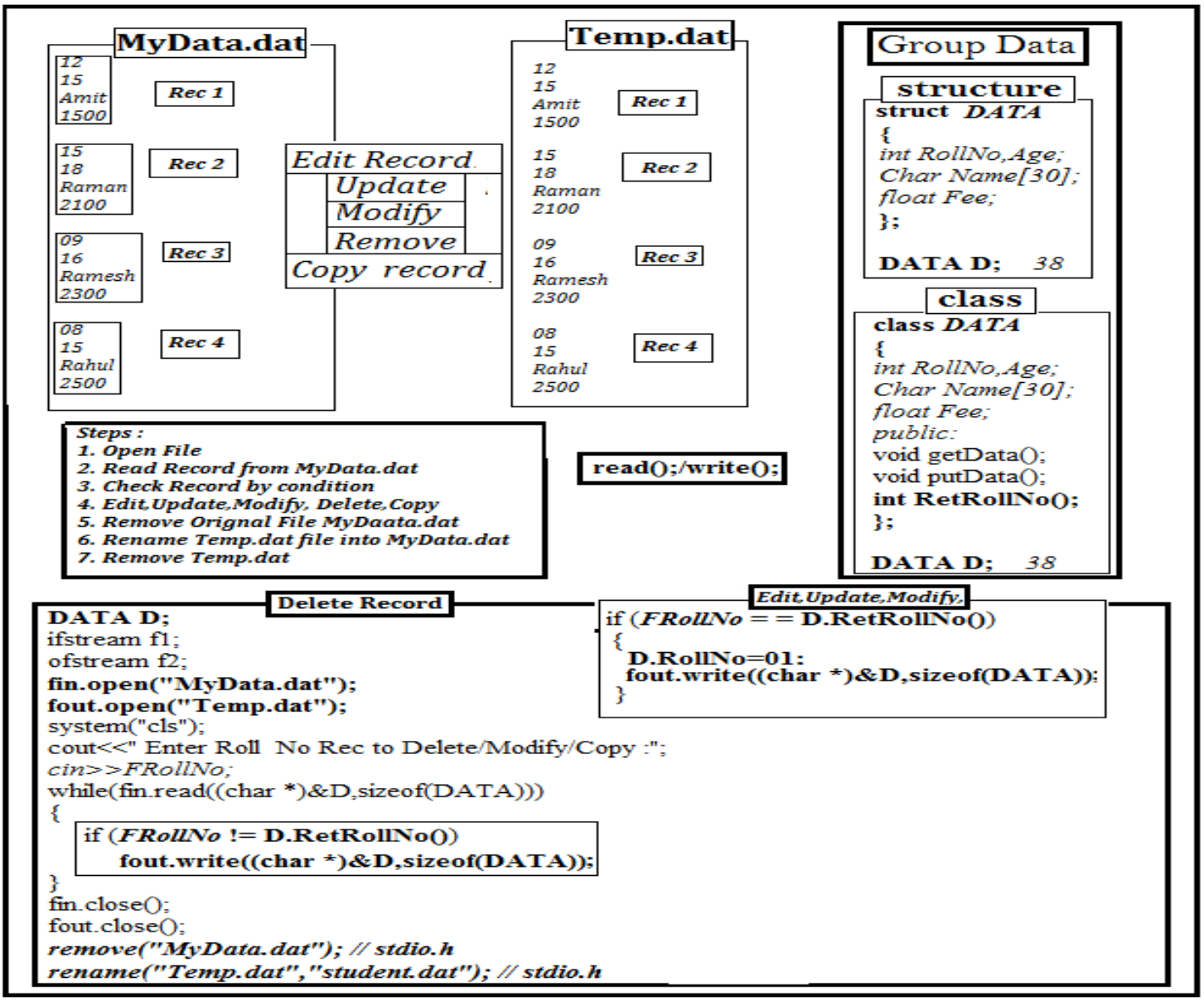
Display a particular Record

structure

```
void main()
{
DATA D;
ifstream fin;
fin.open("MyData.dat");
   int FRollNo;
   cin>>FRollNo;
While( fin.read((char *)&D,sizeof(DATA)))
     {
       if(D.RollNo==FRollNo)
         {
          cout<<"\n Roll Number :"<<D.RollNo;
          cout<<"\n Name :"<<D.Name;
          cout<<"\n Age :"<<D.Age;
          f=1;
          cout<<"\n Fee :"<<D.Fee;
          c++;
          cout<<c;
         }
     }
fin.close();
getch();
}
```

class

```
if(D.RetRollNo() ==FRollNo)
    D.putData();
```



Basic Programs

WAF to read character from file

1)	Character.	**3)**	Lower Case Character	**5)**	Special Character	**7)**	Consonant Character	**9)**	Words
2)	Upper Case Character	**4)**	Digits Character	**6)**	Vowel Character	**8)**	Spaces	**10)**	Lines.

2. WAF to read Words start with from file

a) Words. | **b)** Upper case character | **c)** Lower case character | **d)** Digits character | **e)** Special character

1) Words end with Upper case character

a) Lower Case Character | **b)** Digits Character | **c)** Special Character | **d)** Vowel Character | **e)** Consonant Character

2) Start with vowel and end with consonant character

3) Start with consonant and end with vowel character

4) Start with vowel and consonant present in each word .

5) Words in lines.

3. WAF to read Line from file Words in

a) Lines.

b) Each line.

c) Line start with

d) Upper case character

e) Lower case character

f) Digits character

g) Special character

1) Line end with

a) Upper case character

b) Lower case character

c) Digits character

d) line end with Special character

2) Line start with and end with

a) vowel character | b) consonant character | c) consonant character | d) vowel character

File Handling key Functions

1. get(); char ch=f1.get();
2. put(); char ch=getche(); f1.put(ch);
3. getline(); char str[80]; f1.getline(str,80);
4. tellg(); (ifstream f1;) **[Return the current position of the get pointer]**
   return the current byte position for **reading** a binary file.
   f1.read((char *)&s,sizeof(s));
   f1.read((char *)&s,sizeof(s));
   long pos=f1.tellg();// current position for reading
5. tellp(); (ofstream f1;) **[Return the current position of the put pointer]**
   return the current byte position for **Write** a binary file.
   f1.getdata();
   f1.write((char *)&s,sizeof(s));
   long pos=f1.tellp();// current position of writing pointer
6. seekp(); (ifstream f1;) **[Move get poingter to a specific location]**
   used to place the file pointer to a specified location(byte position) terurn from binary file. ios::beg, ios::curr, ios::end
   f1.seekg(5*sizeof(s));//s object of class/structure and move the pointer to 6th record
   f1.read((char *)&s,sizeof(s));
   f1.display();
7. seekg(); (ofstream f1;) **[Move put poingter to a specific location]**
   used to place the file pointer to a specified location(byte position) terurn from binary file. ios::beg, ios::curr, ios::end
   f1.getdata();
   f1.seekg(5*sizeof(s));//s object of class/structure and move the pointer to 6th record
   f1.write((char *)&s,sizeof(s));

seekg(offset,refposition);

ofset: any constent value and variable (0 to N)
refposition: ios::beg, ios::curr, ios::end
Ex: **f1.seekg(0,ios::beg)** : go to start
f1.seekg(0,ios::cur) : stay at current position
f1.seekg(0,ios::end) : go to end of file
f1.seekg(m,ios::beg) : move to (m+1)th byte in the file
f1.seekg(m,ios::cur) : fo forward by m bytes the current position
f1.seekg(-m,ios::end) : go backward by m bytes from the end

seekp(offset,refposition);

ofset: any constent value and variable (0 to N)
refposition: ios::beg, ios::curr, ios::end
Ex: **f1.seekp(22 ,ios::beg)** : Set the write ponter to the 23r byte (location 22) from the begning of file
f1.seekp(-10,ios::cur) : Set the write ponter to the 11th byte (location 11) from the begning of the file backword from the end of the file
f1.seekp(110 ,ios::end) : Set the write ponter to the 111th byte (location111) from the begning of the file
f1.seekg(5,ios::beg); set the reading pointer to the 6th byte(location 6) the begning of file
f1.seekg(-100,ios::end); Set the read ponter to 101st byte (location 101) backword from the end of the file
f1.seekg(30,ios::cur); Set the read ponter to 31st byte (location 101) from the current position inthe file the end of the file
f1.seekg(0,ios::end); Set the read ponter to the end-of-file

Write Data in File

Q1.

```
#include<iostream.h>
#include<fstream.h>
void main()
{ofstream f1;
f1.opem("Mytext.txt",ios::out);
f1<< "This is my C++ Program for handling text files.";
f1.close();}
```

Q2.

```
#include<iostream.h>
#include<fstream.h>
void main()
{
ofstream f1;
int Age;
char Name[30];
f1.opem("Mytext.txt",ios::out);
cout<< "Enter Your Age and Name: ";
cin>>Age;
gets(Name);
f1<< Name<<" is my friend. He is "<<Age<< " old";
f1.close();}
```

File Operations

a) ifstream-read data from file **[ios::in]**
b) ofstream- weire data in file**[ios::out]**
c) fstream – read and write append**[ios::in|ios::out],[ios::app]**

File Operational steps:

a) Create object for read and write and append
b) Open file for read and write and append using file mods
c) Read and write operation
d) Close file

Open file:

1) For read:

a. ifstream fin("Filename and extention");
b. ifstream fin ;
fin.open ("Filename and extention");
c. ifstream fin ;
fin.open ("Filename and extention",mode);

Mode:- ios::in, ios::app, ios::binary
Example: fin.open("x",ios::in|ios::binary);

2) For write:

a. ofstream fin("Filename and extention");
b. ofstream fin ;
fin.open ("Filename and extention");
c. ofstream fin ;
fin.open ("Filename and extention",mode);

Mode:- ios::out, ios::app, ios::binary
Example: fin.open("x",ios::out|ios::app|ios::binary);

3) For append:

a. fstream fin("Filename and extention");
b. fstream fin ;
fin.open ("Filename and extention");
c. fstream fin ;
fin.open ("Filename and extention",mode);

Mode:- ios::in, ios::out, ios::app, ios::binary
Example:
fin.open("x",ios::out|ios::app|ios::binary);

4) Check file is present or note

```
if(!<file object>)
{
cout<<"Cannot open the file";
exit(1);//process.h
}
```

Open for read

```
if(!fin)
{
cout<<"Cannot open the file";
exit(1);//process.h
}
```

5) Close a file

<file object>.close(); => Example: fin.close();

6) Detect end position:

```
while(!<file object>.eof())
{
//statements
}
```

=>Example:

```
while(!fin.eof())
{
//statements
}
```

7) Write operation in a file

1. I Method:

```
void fwrite()
{
ofstream fout("x",ios::out|ios::binary);
if(!fout)
{
cout<<"Cannot open the file";
exit(1);//process.h}
fout<<"Ravi is my friend\n";
fout<<"He is in my class";
fout.close();
}
```

2. II method:

```
void fwrite()
{
ofstream fout("x",ios::out|ios::binary);
if(fout.is_open())
```

```
{
fout<<"Ravi is my friend\n";
fout<<"He is in my class";
fout.close();
}
else
{
cout<< "Unables to Open";
}
```

3. III method by put() method:

```
void fwrite()
{
char line[80]=" Ravi is my friend\n He is in my class";
ofstream fout("x",ios::out|ios::binary);
if(fout.is_open())
{
for(int i=0;i<strlen(line);i++)
fout.put(line[i]);          //stdio.h
fout.close();
}
else
{
cout<< "Unables to Open";
}
```

8) Reading operation from a file:

1. I Method reading using get():

```
void fread()
{
ifstream fin("x",ios::in|ios::binary);
char c;
if(!fin)
{
cout<<"Cannot open the file";
exit(1);//process.h
}
else
{
while(fin)
{
fin.get(c);
cout<<c;
}
fin.close();
}
```

2. II method read line using getline():

```
void fread ()
{
ifstream fin("x",ios::in|ios::binary);
char line[80];
if(!fin)
{
cout<<"Cannot open the file";
exit(1);//process.h
}
else
{
while(fin)
{
getline (fin,line);
cout<<line;
}
fin.close();
}
```

9) Append into existing file:

```
void fwrite()
{
char line[80]=" Ravi is my friend\n He is in my class";
ofstream fout("x",ios::app|ios::binary);
if(fout.is_open())
{
for(int i=0;i<strlen(line);i++)
fout.put(line[i]);          //stdio.h
fout.close();
}
else
{
cout<< "Unables to Open";
}
```

Binary File read and write methods used on grouped data (structure and class)

Syntax: read(memory,size);

Example :

For Structure: fin.read((char*)&[structure Object],sizeof([Structure Name]);

For Class : fin.read((char*)&[class Object],sizeof([class Object]);

Fileobject.read((char *)&object, sizeof(object));

Syntax: write(memory,size);

Example :

For Structure: fin.write((char*)&[structure Object],sizeof([Structure Name]);

For Class : fin.write((char*)&[class Object],sizeof([class Object]);

Fileobject.write((char *)&object, sizeof(object));

Example Programs

1. Example of write ( ) member function Structure

```
#include<fstream>
#include<iostream>
using namespace std;
struct student
{
```

```
int roll ;
char name[30];
char address[60];
};
void main()
{
student s;
ofstream fout;
fout.open("student.dat");
cout<<"\n Enter Roll Number :";
cin>>s.roll;
cout<<"\n Enter Name :";
cin>>s.name;
cout<<"\n Enter address :";
cin>>s.address;
fout.write((char *)&s,sizeof(student));
fout.close();
getch();
}
```

2. To Read data from a binary File using read( ) member function Structure

```
#include<fstream>
#include<iostream>
#include<conio.h>
using namespace std;
struct student
{ int roll ;
char name[30];
char address[60];
};
void  main()
{
DATA D;
ifstream fin;
fin.open("student.dat");
fin.read((char *)&s,sizeof(student));
cout<<"\n Roll Number :"<<s.roll;
cout<<"\n Name :"<<s.name;
cout<<"\n Address :"<<s.address;
fin.close();
getch();
}
```

3. Writing Class object in a file

```
#include<fstream>
#include<iostream>
using namespace std;
class student
{
int roll ;
char name[30];
char address[60];
public:
void read_data( ); // member function prototype
void write_data( ); // member function prototype
};
void student::read_data( ) // member function defintion
{
 cout<<"\n Enter Roll :";
cin>>roll;
cout<<"\n Student name :";
cin>>name;
cout<<"\n Enter Address :";
cin>>address;
}
void student:: write_data()
{
cout<<"\n Roll :"<<roll;
cout<<"\n Name :"<<name;
cout<<"\n Address :"<<address;
}
int main()
{
student s;
ofstream fout;
fout.open("student.dat");
s.read_data(); // member function call to get data from KBD
fout.write((char *)&s,sizeof(student)); // write object in file
fout.close();
return 0;
}
```

4. Reading Class object from a binary file

```
#include<fstream>
#include<iostream>
#include<conio.h>
using namespace std;
class student
{
 int roll ;
char name[30];
char address[60];
public:
void read_data( ); // member function prototype
void write_data( ); // member function prototype
};
void student::read_data( ) // member function definition
{ cout<<"\n Enter Roll :";
cin>>roll;
cout<<"\n Student name :";
cin>>name;
cout<<"\n Enter Address :";
cin>>address;
}
void student:: write_data()
{
cout<<"\n Roll :"<<roll;
cout<<"\n Name :"<<name;
cout<<"\n Address :"<<address;
}
int main()
```

```
{
 student s;
ifstream fin;
fin.open("student.dat");
fin.read((char *)&s,sizeof(student));
s.write_data();
fin.close();
getch();
return 0;
}
```

10) Some other very important member function:

Member function name	Explanation

seekg( )	Used to move **reading** pointer forward and backward
	Syntax fileobject.seekg(no_of_bytes,mode);
	Example: (a) fout.seekg(50**,ios::cur**); // move 50 bytes forward from current position
	(b) fout.seekg(50,**ios::beg**); // move 50 bytes forward from current beginning
	(c) fout.seekg(50,**ios::end**); // move 50 bytes forward from end .

seekp( )	Used to move **writing** pointer forward and backward
	Syntax **fileobject.seekp(no_of_bytes,mode);**
	Example: (a) fout.seekp(50,**ios::cur**); // move 50 bytes forward from current position
	(b) fout.seekp(50,**ios::beg**); // move 50 bytes forward from current beginning
	(c) fout.seekp(50,**ios::end**); // move 50 bytes forward from end .

tellp( )	It return the distance of **writing** pointer from the beginning in bytes Syntax Fileobject.tellp( ); Example: long n = fout.tellp( );
tellg( )	It return the distance of **reading** pointer from the beginning in bytes Syntax Fileobject.tellg( ); Example: long n = fout.tellg( );

Files MODES:

File mode	Explanation
ios::in	Input mode – Default mode with **ifstream** and files can be read only
ios::out	Output mode- Default with **ofstream** and files can be write only
ios::binary	Open file as binary
ios::app	Preserve previous contents and write data at the end (move forward only)
ios::ate	Preserve previous contents and write data at the end.(can move forward and backward)
ios::nodelete	Do not delete existing file
ios::noreplace	Do not replace file
ios::nocreate	Do not create file

NOTE : To add more than one mode in a file stream use bitwise OR (|) operator

Difference and Definition

Text Files	**Binary Files**
In these types of files all the data is firstly converted into their equivalent char and then it is stored in the files.	In these types of files all the data is stored in the binary format as it is stored by the operating system. So no conversion takes place. Hence the processing speed is much more than text files.

get( ) member function	**getline()** function

Get() function is used to read a single char from the input stream in text file
Syntax
fileobject.get(char);
Example:
fin.get(ch); //fin is file stream.
	Getline() function is used to read a string from the input stream in text file.
	Syntax
	fileobject.getline (string, no_of_char,delimiter);
	Example
	fin.getline(str,80); // fin is file stream.
	NOTE: Delimiter is optiona

2/3 Marks Practice Questions

1. Write a function in C++ to count the number of uppercase alphabets present in a text file "BOOK.txt"
2. Write a function in C++ to count the number of alphabets present in a text file "BOOK.txt"
3. Write a function in C++ to count the number of digits present in a text file "BOOK.txt"
4. Write a function in C++ to count the number of white spaces present in a text file "BOOK.txt"
5. Write a function in C++ to count the number of vowels present in a text file "BOOK.txt"
6. Assume a text file "Test.txt" is already created. Using this file, write a function to create three files "LOWER.TXT" which contains all the lowercase vowels and "UPPER.TXT" which contains all the uppercase vowels and "DIGIT.TXT" which contains all digits.

3 Marks Solve Question

1. Write a function in c++ to search for details (Phoneno and Calls) of those Phones which have more than 800 calls from binary file "phones.dat". Assuming that this binary file contains records/ objects of class Phone, which is defined below.

```
class Phone
{
Char Phoneno[10]; int Calls;
public:
void Get() {gets(Phoneno); cin>>Calls;}
void Billing() { cout<<Phoneno<< "#"<<Calls<<endl;}
int GetCalls() {return Calls;}
};
```

Ans 1 :

```
void Search()
{
Phone P;
fstream fin;
fin.open( "Phone.dat", ios::binary| ios::in);
while(fin.read((char *)&P, sizeof(P)))
{
if(p.GetCalls() >800)
p.Billing();
}
Fin.close(); //ignore
}};
```

2. Write a function in C++ to add new objects at the bottom of a binary file "STUDENT.DAT", assuming the binary file is containing the objects of the following

```
class STUD
{int Rno;
char Name[20];
public:
void Enter()
{cin>>Rno;gets(Name);}
void Display(){cout<<Rno<<Name<<endl;}
};
```

Ans.2.

```
void searchbook(int bookno)
{ifstream ifile("BOOK.DAT",ios::in|ios::binary);
if(!ifile)
{cout<<"could not open BOOK.DAT file"; exit(-1);}
else
{BOOK b; int found=0;
while(ifile.read((char *)&b, sizeof(b)))
{if(b.RBno()==bookno)
{b.Display(); found=1; break;}
}
if(! found)
cout<<"record is not found ";
ifile.close();
}
}
```

3. Given a binary file PHONE.DAT, containing records of the following class type class Phonlist

```
{
char name[20];
char address[30];
char areacode[5];
char Phoneno[15];
public:
void Register()
void Show();
void CheckCode(char AC[])
{return(strcmp(areacode,AC);
}};
```

Write a function TRANSFER( ) in C++, that would copy all those records which are having areacode as "DEL" from PHONE.DAT to PHONBACK.DAT.

Ans 3.

```
void TRANSFER()
{
fstream File1,File2;
Phonelist P;
File1.open("PHONE.DAT", ios::binary|ios::in);
File2.open("PHONEBACK.DAT", ios::binary|ios::OUT)
while(File1.read((char *)&P, sizeof(P)))
{ if( p.CheckCode( "DEL"))
File2.write((char *)&P,sizeof(P)); }
File1.close();
File2.close();
}
```

Program to explain the different operation for Project:

```
#include<iostream>
#include<fstream>
#include<conio.h>
using namespace std;
class student
{ int admno;
char name[30];
char address[60];
public:
void read_data()
{
 cout<<"\n Enter Admission No :";
cin>>admno;
fflush(stdin);
cout<<"\n Enter Name :";
cin.getline(name,29);
fflush(stdin);
cout<<"\n Enter Address :";
cin.getline(address,59);
}
void write_data()
{ cout<<"\n\n Admission No
:"<<admno;
cout<<"\n Name :"<<name;
cout<<"\n Address :"<<address;
}
int get_admno()
{
return admno;
}
};
void write_to_file(void)
{ student s;
ofstream fout;
fout.open("student.dat",ios::app);
s.read_data();
fout.write((char *)&s,sizeof(student));
fout.close();
return;
}
void read_from_file()
{ student s;
ifstream fin;
fin.open("student.dat");
while(fin.read((char
*)&s,sizeof(student)))
s.write_data();
fin.close();
return;
}
// function to modify student
information
void modify_record(void)
{
 int temp_admno;
student s;
ifstream fin;
ofstream fout;
fin.open("student.dat");
fout.open("temp.dat");
system("cls"); // header file stdlib.h
cout<<"\n Enter admission No to
Modify :";
cin>>temp_admno;
while(fin.read((char
*)&s,sizeof(student)))
{ if (temp_admno==s.get_admno())
{
s.read_data();
}
fout.write((char *)&s,sizeof(student));
}
fin.close();
fout.close();
remove("student.dat");
rename("temp.dat","student.dat");
return;
}
void modify_alternate_method()
{
 student s;
int temp_admno;
fstream file;
file.open("student.dat",ios::in|ios::out|io
s::ate|ios::binary);
cout<<"\n Enter admno to modify :";
cin>>temp_admno;
file.seekg(0); // one method to reach at
begining
// long n = file.tellg(); // find out total
no of bytes
// file.seekg((-1)*n,ios::end); // move
backward total no of bytes from end
while(file.read((char*)&s,sizeof(student
)))
{ if(temp_admno == s.get_admno())
{ s.read_data();
int n = -1*sizeof(student);
file.seekp(n,ios::cur);
file.write((char *)&s,sizeof(student));
}
}
file.close();
return;
}
void delete_record(void)
{
 int temp_admno;
student s;
ifstream fin;
ofstream fout;
fin.open("student.dat");
fout.open("temp.dat");
system("cls");
cout<<"\n Enter admission No to Delete
:";
cin>>temp_admno;
while(fin.read((char
*)&s,sizeof(student)))
{
if (temp_admno!=s.get_admno())
fout.write((char *)&s,sizeof(student));
}
fin.close();
fout.close();
remove("student.dat"); // stdio.h
rename("temp.dat","student.dat"); //
stdio.h
return;
}
void search_record()
{
 int found=0;
student s;
int temp_admno;
ifstream fin("student.dat");
cout<<"\n Enter Admno to search :";
cin>>temp_admno;
while(fin.read((char*)&s,sizeof(student
)))
{
if(temp_admno==s.get_admno())
{
found=1;
s.write_data();
}
}
fin.close();
if(found ==0)
cout<<"\n Admission No.
"<<temp_admno<<" does not exist ";
getch();
return;
}
void count_record(void)
{
 int count=0;
student s;
int temp_admno;
ifstream fin("student.dat");
while(fin.read((char*)&s,sizeof(student
)))
count++;
fin.close();
cout<<"\n Total Record :"<<count;
getch();
return ;
}
int main()
{ int choice;
do
{
system("cls"); // stdlib.h
cout<<"\n\n\t\t\t MAIN MENU ";
cout<<"\n\t\t\t\t1. Add Student ";
cout<<"\n\t\t\t\t2. Show Student";
cout<<"\n\t\t\t\t3. Modify Record";
cout<<"\n\t\t\t\t4. Modify Record
(Alternate Method)";
cout<<"\n\t\t\t\t5. Delete Record";
```

```
cout<<"\n\t\t\t\t6. Count Record";
cout<<"\n\t\t\t\t7. Search Record";
cout<<"\n\t\t\t\t8. Exit";
cout<<"\n\n\t\t\t Enter your choice :";
cin>>choice;
switch(choice)
{
case 1: system("cls");
write_to_file();
break;
case 2: read_from_file();
getch();
break;
case 3: modify_record();
break;
case 4: modify_alternate_method();
break;
case 5: delete_record();
break;
case 6: count_record();
break;
case 7: search_record();
break;
case 8: break;
default: cout<<"\n Wrong choice.... Try again";
getch();
}
}while(choice!=8);
return 0;
}
```

Solved Questions

1. Write a function in a C++ to read the content of a text file "DELHI.TXT" and display all those lines on screen, which are either starting with 'D' or starting with 'M'. [CBSE 2012]

```
void DispDorM()
{
ifstream File("DELHI.TXT")
char str[80];
while(File.getline(str,80))
{
if(str[0] = ='D' || str[0] = ='M')
cout<<str<<endl;
}
File.close(); //Ignore
}
```

2. Write a function in a C++ to count the number of lowercase alphabets present in a text file "BOOK.txt".

```
int countalpha()
{ ifstream Fin("BOOK.txt");
char ch;
int count=0;
while(!Fin.eof())
{
Fin.get(ch);
if (islower(ch))
count++;
}
Fin.close();
return count;
}
```

3. Function to calculate the average word size of a text file.

```
void calculate()
{ fstream File;
File.open("book.txt",ios::in);
char a[20];
char ch;
int i=0,sum=0,n=0;
while(File)
{ File.get(ch);
a[i]=ch;
i++;
if((ch==' ') || ch(== '.')||(char==',')(ch=='\t')||(ch=='\n')
{i --; sum=sum +i;
i=0; N++;
}
} cout<<"average word size is "<<(sum/n);
}
```

4. Assume a text file "coordinate.txt" is already created. Using this file create a C++ function to count the number of words having first character capital.

```
int countword()
{ ifstream Fin("BOOK.txt");
char ch[25];
int count=0;
while(!Fin.eof())
{Fin>>ch;
if (isupper(ch[0]))
count++;
}
Fin.close();
return count;
}
```

5. Function to count number of lines from a text files (a line can have maximum 70 characters or ends at '.')

```
int countword()
{ ifstream Fin("BOOK.txt");
char ch[70];
int count=0;
if (!Fin)
{ cout<<"Error opening file!" ;
exit(0);
}
while(1)
{Fin.getline(ch,70,'.');
if (Fin.eof())
break;
count++;
}
Fin.close();
return count;
}
```

Problem 1:	Given the binary file STUDENT.DAT , containing the records of the following class:
	class student
	{
	int roll_no;
	char name[20];
	float percent;
	public:
	void getData( );
	void show( );
	float returnPercent( )
	{return percent;
	}
	};
	Write a function BELOW75( ) in C++ , that would count and display the records of those students whose score is below 75 percent.

Problem 2:	class book
	{int book_no;
	char book_name[20];
	float price;
	public:
	void enter_book_Details( )
	{
	cin>> book_no>> price; gets(book_name);
	}
	void show_book_Details( );
	};
	Assuming a binary file "BOOK.DAT" contains objects belonging to class book, write a user-defined function to add more records to the end of it.

Problem 3:	Write a function in C++ to count and display the number of student records stored in the binary file "Student,dat" . Assume that student is a structure and 10 bytes of memory is required to store each student record.
Problem 4:	Write a function in C++ to count the number of uppercase alphabets present in a text file "STORY.TXT".
Problem 5:	Write a function in C++ to count the number of alphabets present in a text file "XY.TXT".
Problem 6:	Write a function in C++ to count and display the number of lines starting with alphabet „A" in a text file "MYFILE.TXT".
Problem 7:	Write a function in C++ to count the number of words present in the text file "MyFile.txt". Assume that each word is separated by a blank space and no blank space appears in the beginning and at the end of the file.
Problem 8:	26 A librarian maintains the record of books in a file named as "STOCK_BOOK.DAT". Write a function in C++ to delete a record for book_no 10.
Problem 9:	Given the binary file TELEPHONE.DAT , containing the records of the following class Directory: class Directory { char name[20]; char address[30]; char areaCode[5]; char phone_no[15];

	public:
	void register( );
	void show( );
	int checkCode(char AC[])
	{
	return strcmp(areaCode, AC);
	}
	};
	Write a function COPYABC( ) in C++ , that would copy only those records having areaCode as"123" from TELEPHONE.DAT to TELEBACK.DAT.

Problem 10:	Write a function in C++ to count the number of vowels present in a text file STORY.TXT".

Problem 11:	Observe the program segment carefully and answer the question that follows:
	class item
	{
	int item_no;
	char item_name[20];
	public:
	void enterDetail( );
	void showDetail( );
	int getItem_no( ){ return item_no;
	}
	};
	void modify(item x)
	{
	fstream File;
	File.open("item.dat", ios::binary|ios::in|ios::out) ;
	item i;
	while(File .read((char*) & i , sizeof (i)))//Statement 1
	{
	if(x . getItem_no( ) = = i . getItem_no( ))
	{
	File.seekp(File.tellg( ) – sizeof(i));
	File.write((char*) &x , sizeof (x));
	}
	}
	File.close() ;
	}
	If the function modify( ) modifies a record in the file " item.dat" with the values of item x passed as argument, rewrite statement 1 in the above code using ios::eof( ) , so as to modify record at its proper place.

Problem 12:	A file named as "STUDENT.DAT" contains the student records, i.e. objects of class student. Assuming that the file is just opened through the object FILE of fstream class, in the required file mode, write the command to position the get pointer to point to fifth record from the beginning.

FILE HANDLING

Q.1. How many stream objects would you need to manage the following situations:
(i) To process two files one after other
(ii) To merge two sorted files into third file?

Q.2. Define Stream? Explain input stream and output stream?

Q.3. Differentiate between
i) get() and getline() function
ii) read() and write() functions
iii) write() and put()
iv)Binary file and text file
v) ios::noreplace and ios::nocreate
vi) tellg() and seekg()
vii) bad() and good() function

Q.4. Why all the functions of ostream class can be called in fstream class?

Q.5. Name the base class of the stream class hierarchy?

Q.6. Name the stream classes supported by C++ for file input and output.

Q.7. Distinguish between ios::out and ios::app

Q.8. Find errors if any:

i)
```
ifstreaminfile;
infile.open("dos.txt",ios.app);
```

ii)
```
ifstream. infile;
infile.close("dos.txt");
```

iii)
```
ofstream f;
whileeof
cout<<"End of file";
```

iv)
```
intbuf[100], n=10;
cin.getline(buf,10,'\n');
```

v)
```
ifstreaminf;
ofstreamoutf;
inf>>Name;
outf>>class;
```

vi)
```
char *buf;
cout.write(buf).put('\n').put('\n');
```

Q.9. How do you detect end of file? Also give suitable lines of code to find whether file is opened successfully or not.

Q.10. Describe different file modes of used in file operations?

Q.11.Consider the following class declaration:

```
class employee
{
        int code;
        char name[20];
        float salary ;
public:
        void input() { cin>>name>>salary;}
        void show() { cout<<code<<name<<salary<<endl;}
        floatretsal() { return salary;}
};
```

Give function definitions to do the following:

i) Write the objects of employee to a binary file.

ii) Read the object of employee from binary file and display all the objects on the screen where salary is between 10,000 to 20,000.

Q.12. Write suitable functions:

i) Put the get pointers at specified offset.

ii) Get the position of put pointer.

iii) Detect the end of file.

iv)Check whether input output operation has failed.

Q.13. A data file contains name and telephone number as two fields. Write a menu driven program to do the following:

i) Add record in the file. ii) Searching for a telephone number for a given name.

iii) Update the data file to make changes in the telephone number.

Q.14. Write a program to search for specified record whose record number is given by the user in a given file " item.dat" having itemno, name, uprice as fields.

Q.15. Given a binary file **APPLY.DAT,** containing records of the following class Applicant .

```
class Applicant
{
        charA_Rno[10];                  // Roll number of applicant
        charA_Name[30];                 // Name of applicant
        intA_Score;                     // Score of applicant
public:
        voidEnrol()
        {                       gets(A_Rno) ; gets (A_Name) ; cin>>A_Score;   }

        void Status()
        { cout<<setw(12)<<A_Admno<<setw(32)<<A_Name<<setw(3)<<A_Score<<endl;
        }
        intReturnScore() {return A_Score; }
};
```

Write a function in C++ that would read contents of file APPLY.DAT and display the details of those students who have scored distinction marks and marks below 50 under appropriate headings.

Q.16.Consider the following class declaration:

```
Class FLIGHT
{
        intflight_no;
        char destination[20];
        float distance;
public:
        void INPUT();  // reads object
        voidWrite_File(); // write record to the file
        void OUTPUT(); // displays object
};
```

Complete the member functions definition.

17. Observe the program segment carefully and answer the question that follows:

```
class member
{
intmember_no;
```

```
charmember_name[20];
public:
voidenterDetails( );
voidshowDetail( );
intgetMember_no( ){ return member_no;}
};
void update(member NEW )
{
fstream File;
File.open( "member.dat", ios::binary|ios::in|ios::out) ;
member i;
while(----------)
{
if(NEW . getMember_no( ) = = i . getMember_no( ))
{
File.seekp( _____ , ios::cur ) //Paremeter Missing
File.write((char*) &NEW , sizeof (NEW));
}}
File.close() ;}
```

If the function update( ) is supposed to modify a record in the file "member.dat" with the values of member NEW passed as argument, write the appropriate missing statements.

18 A file named as "STUDENT.DAT" contains the student records, i.e. objects of class student. Assuming that the file is just opened through the object FILE of fstream class, in the required file mode, write the command to position the put pointer to point to second record from the end of file, read new record and rewrite at the set location.

19. Read the code given below and answer the question:

```
void main( )
{
charch = 'A';
fstreamoutFile ("data.dat", ios::out);
outFile<<ch<<ch;
}
```

If the file contains "GOOD" before execution, what will be the contents of the file after execution of this code?

20. Write a function in C++ to count and display the number of student records stored in the binary file "Student,dat" . Assume that student is a structure and 10 bytes of memory is required to store each student record.

21. Given the binary file STUDENT.DAT, containing the records of the following class:

```
class student
{  introll_no;
char name[20];
float percent;
public:
voidgetData( );
void show( );
floatreturnPercent( ){ return percent; }
};
```

I. Write a function BELOW75( ) in C++ , that would count and display the records of those
II. students whose score is below 75 percent.
III. Write a function update() to modify record for students having rollno n .
IV. Write a function Delrec() to delete a record for student having rollno n.

DATA FILE HANDLING IN C++

Set1

Q1. Observe the program segment given below carefully and fill in the blanks marked in statement1 and statement2 using seekp() and seekg() functions for performing the required task.

```
class Item
{
int Ino;char Item[20];
public:
//Function to search and display the content from a
particular record number
void Search(int);
// Function to modify the content from a particular
record number
void Modify(int);
};
void Item:: Search(int RecNo)
{
fstream File;
File.open("STOCK.DAT",ios::binary|ios::in);
_____ // statement1
file.read((char*)this,sizeof(Item));
cout<<Ino<<"□"<<Item<<endl;
File.claos();
}
void Item::Modify(int RecNo)
{
fstream File;
File.open("STOCK.DAT",ios::binary|ios::in|ios::out);
cout>>Ino;cin.getline(Item,20);
_____ ??Statment 2
File.write((char*)this,sizeof(Item));
File.close();
}
```

Answer:

File.seekg(RecNo*sizeof(Item)); //Statement 1

File.seekp(RecNo*sizeof(Item)); //Statement 2

Q2. Observe the program segment given below carefully and fill in the blank marked in Statement 1 and Statement 3 using seekg() and tellg() functions for performing the required task:

```
class Employee
{
int Eno;char ename[20];
public:
//Function to count the total number of records
int Countrec();
};
int Item::Countrec()
{
fstream File
File.open("EMP.DAT",ios::binary|ios::in);
_____ // statement 1
int Bytes=_____ //statement 2
int Count = Bytes / sizeof(Item);
File.close();
return Count;
}
```

Answer:-

File.seekg(0,ios::end); //statement 1

File.tellg(); //statement 2

Q3. Observe the program segment given below carefully, and answer the question that follows : 1

```
class Labrecord
{
int Expno;
char Experiment[20];
char Checked;
int Marks;
public :
//function to enter Experiment details
void EnterExp( );
//function to display Experiment details
void ShowExp ( ) ;
//function to return Expno
char RChecked ( ) {return Checked;
}
//function to assign Marks
void Assignmarks(int M)
{
Marks = M;
}
};
void MpdifyMarks()
{
fstream File;
File.open("Marks.Dat",ios::binary|ios::in|ios::out);
Labrecord L;
int Rec = 0;
while (File.read((char*)&L, sizeof(L)))
{
if(L.RChecked( )== ' N ' )
L.Assignmarks(0)
else
L.Assignmarks(10)
_____ //statement 1
_____ //statement 2
Rec ++ ;
}
File.close ();
}
```

If the funption ModifyMarks() is supposed to modify Marks for the records in the file MARKS.DAT based on

their status of the member Checked (containing value either V or 'N'). Write C++ statements for the statement 1 and statement 2, where, statement 1 is required to position the file write pointer to an appropriate place in the file and statement 2 is to perform the write operation with the modified record.

Answer :

(a) Statement 1:

File.seekg(-1*sizeof(L),ios::cur);

OR

File.seekg(Rec*sizeof(L));

OR

File.seekp(-1*sizeof(L),ios::cur);

OR

File.seekp(Rec*sizeof(L));

OR

Any equivalent correct method of calculating size of the record in place of sizeof operator.

Statement 2:

File.write((char *) &L,sizeof(L));

OR

Any equivalent correct method of calculating size of the record in place of sizeof operator.

Q4. Observe the program segment given below carefully, and answer the question that follows: 1

```
class PracFile
{       intPracno;
char PracName[20];
int TimeTaken;
int Marks;
public:
// function to enter PracFile details
void EnterPrac( );
// function to display PracFile details
void ShowPrac( ):
// function to return TimeTaken
int RTime() {return TimeTaken;}
// function to assign Marks
void Assignmarks (int M)
{ Marks = M;}
};
void AllocateMarks( )
{ fstreamFile;
File.open("MARKS.DAT",ios::binary|ios::in|ios::out);
PracFile P;
int Record = 0;
while (File.read(( char*) &P, sizeof(P)))
{       if(P.RTime()>50)
P.Assignmarks(0)
else
P.Assignmarks(10)
_____ //statement 1
_____ //statement 2
Record + + ;
}
File.close();
}
```

If the function AllocateMarks () is supposed to Allocate Marks for the records in the file MARKS.DAT based on their value of the member TimeTaken.

Write C++ statements for the statement 1 and statement 2, where, statement 1 is required to position the file write pointer to an appropriate place in the file and statement 2 is to perform the write operation with the modified record.

Answer:-

(a) Statement 1:

File.seekp(Record * sizeof(P));

OR

File.seekp(Record * sizeof(PracFile));

OR

File.seekp(-sizeof(P), ios::cur);

OR

File.seekg(Record * sizeof(P));

OR

File.seekg(Record * sizeof(PracFile));

OR

File.seekg(-sizeof(P), ios::cur);

OR

Any equivalent correct method of calculating size of the record in place of sizeof operator.

Statement 2:

File.write((char*)&P, sizeof(P));

OR

File.write((char*)&P, sizeof(PracFile));

OR

Any equivalent correct method of calculating size of the record in place of sizeof operator.

Q5. Observe the program segment given below carefully ,and answer the question that follows:- 1

```
class Book;
{ int Book_no;
```

```
 char Book_name[20];
public:
//function to enter Book details
void enterdetails()
//function to display Book details
void showdetails();
//function to return Book_no
int Rbook_no(){return Book_no;}
};
void Modify (Book NEW)
{
fstream File;
File.open("BOOK.DAT",ios::binary|ios::in|ios::out);
Book OB;
int Recordsread=0 ,Found=0;
while(!Found && File.read((char*)&OB,sizeof(OB)))
{
Recordsread++;
if(NEW.RBook_no()==OB.RBook_no())
{
_____  //Missing Statement
File.write((char*)&NEW,sizeof(NEW));
Found=1;
}
else
File.write((char*)&OB,sizeof(OB);
}
if (!Found)
cout<<"Record for modification does not exist";
File.close();
}
```

If the function Modify() is supposed to modify a record in file BOOK.DAT with the value of Book NEW passed to its argument ,write the appropriate statement for missing statemet using seekp() or seekg() ,whichever needed ,in the above code that would write the modicfied record at its proper place.

Answer:-

File.seekg(-1*sizeof(NEW),ios::out);

Q6. S "student.dat" file exist,with the object of class students .Assuming ,the file has just been opened through the object fil of fstream class Give a single command to place the file pointer to the third record from beginning.

In continuation to the above, command ,give a command to bring file pointer to the beginning of last record.

Answer:-

fil.seekg(3*sixeof(student),ios::beg);
fil.seekg(sizeof(student),ios::end);

DATA FILE HANDLING IN C++ Set2

02 Marks Questions

Q1. Write s c++ program, which reads one line at a time from the disk file TEST.TXT and display it to a monitor. Your program has to read all the contents of the file. Assume the length of the line not to exceed 80 characters .You have to include all the header files if required.

Answer:-

```
#include <iostream.h>
#include <stdlib.h>
#include <fstream.h>
ifstream in_file;
main( )
{
char in_char;
char line[80];
in_file.open("TEST.TXT", ios::in);
if(!in_file)
{
cerr << "\n\n File does not exist";
exit(0);
}
while(in_file.getline(line,80))
{
cout << line << endl;
}
in_file.close( );
return 0;
}
```

Q2. Write a c++ program, which initialize a string variable to the content."Time is a great teacher but unfortunately it kills all its pupils.Berlioz" and outputs the string one character at a time to the disk file OUT.TXT.You have to include all the header files if required.

Answer:

```
#include <fstream.h>
void main( )
```

```
{
char line[80] = {"Time is a great teacher but
unfortunately it kills all its pupils."};
ofstream fout("OUT.TXT");
for (int C=0; line[C] != '\0'; C++)
fout << line[C];
fout << endl;
fout.close( );
}
```

Q3. Assuming a binary file JOKES.DAT is containing objects belonging to a class JOKE(as defined below).Write a user defined function in c++ to add more objects belonging to class JOKE at the bottom of it.

```
class JOKES
{
int Jokied;         //Joke Identification number
char Type[5];//Joke Type
char Jokedesc[255]; //Joke Description
public:
void Newjoykeentry()
{
cin>>Jokeid;
gets(Type);
gets(Jokedesc);
}
void Showjoke()
{
cout<<Jokeid<<" "<<Type<<endl<<Jokedesc<<endl;
}
};
```

Answer:-

```
 // Function to add more objects belonging to class JOKE
at the bottom of JOKES.DAT
void append()
{
fstream afile;
afile.open("JOKES.DAT", ios::binary|ios::app);
JOKE  LAUG;
int n, i;
cout << "How many objects you want to add : ";
cin >> n;
for(i = 0; i<n; i++)
{
LAUG.Newjokeentry();
afile.write((char *)&LAUG, sizeof(JOKE));
}
afile.close();
}
```

Q4. Write a user defined function in c++ to read the content from a text file NOTES.TXT ,count and display the number of blank spaces present in it.

Answer:-

```
#include <fstream.h>
#include <iostream.h>
#include <ctype.h>
#include <conio.h>
void display()
{
ifstream afile;
/* If NOTES.txt contains the following line :
C++ File handing in Class-12 */
afile.open("NOTES.TXT");
char ch;
int c = 0;
while(afile)
{
afile.get(ch);
if (ch == ' ' )
c++;
}
cout << "The number of blank spaces : " << c;
}
void main()
{
clrscr();
display();
}
```

Q5. Assume that a text file named TEXT.TXT already contains some text written into it ,write a function named vowelwords(),that reads the file TEXT1.TXT and create a new file named TEXT2.TXT ,which shall contain only those words from the file TEXT1.TXT which don't star with an uppercase vowel(i.e. with 'A','E','I','O','U') ,for example if the file TEXT1.TXT contains

Carry Umbrella and Overcoat When it Rains

then the file TEXT2.TXT shall contain

Carry when it Rains.

Answer:-

```
void vowelwords()
{
fstream afile,bfile;
char ch,ch1;
afile.open("TEXT1.TXT", ios::in);
bfile.open("TEXT2.TXT", ios::out);
ch1 = ' ';
clrscr();
while(afile)
```

```
{
afile.get(ch);
cout << "\nOutside " << ch;
if (( ch =='A') || (ch =='E') ||
(ch=='I')||(ch=='O')||(ch=='U')&&(ch1==' '))
{
while(ch != ' ')
{
afile.get(ch);
ch1 = ch;
if(ch == ' ')
break;
}
}
else
bfile.put(ch);
}
afile.close();
bfile.close();
}
```

Q6. Write a function in c++ to count the number of lines present in a text file "STORY.TXT".

Answer:-

```
void CountLine()
{
ifstream FIL("STORY.TXT");
int LINES = 0;
char STR;
while (FIL)
{
FIL.get(STR);
if (STR == '.')
LINES++;
}
cout << "No. of Lines : " << LINES << endl;
}
```

Q7. Write a function in c++ to count the number of alphabets present in a text file "NOTES.TXT".

Answer:

```
void CountAlphabet()
{
ifstream FIL("NOTE.TXT");
int CALPHA = 0;
char CH = FIL.get();
while (!FIL.eof())
{
if (isalpha(CH))
CALPHA++;
CH = FIL.get();
}
cout << "No. of Alphabets : " << CALPHA << endl;
}
```

Q8. Write a function to calculate the average word size in a text file "Report.txt" each word is separated by a single space or full stop.

Answer:

```
void calculate()
{
fstream tfile;
clrscr();
tfile.open("Report.txt",ios::in);
char arr[20];
char ch;
int i=0,sum=0,n=0;
while(tfile)
{
tfile.get(ch);
arr[i] = ch;
i++;
if (( ch == ' ') || (ch == '.'))
{
i--;
sum = sum + i;
i = 0;
n++;
}
}
cout << " Average word size is  "<<(sum/n);
}
```

03

Q1. Assuming the class Employee given below, write function in c++ to perform following:-

Write the object of Employee to a binary file.

Read the objects of employee from binary file and display them on screen.

```
class employee
{
int eno;
char ename;
public:
void getit()
{
cin>>eno;
gets(ename);
}
void showit()
{
```

```
cout<<eno;
cout<<ename;
}
}
```

Answer:-

(a)

```
// Function to write the object of class into binary file
void Write_file()
{
Employee emp;
ofstream efile;
efile.open("emp.dat", ios :: binary || ios :: out);
int n, i;
cout << "Enter how many employee : ";
cin >> n;
for (i =0; i<n; i++)
{
emp.getit();
efile.write((char *)& emp, sizeof(emp));
}
}
```

(b)

```
//Function to display the object of class
void display()
{
Employee emp;
ifstream efile;
efile.open("emp.dat", ios :: binary || ios :: in);
while(efile)
{
efile.read((char *)&emp , sizeof(emp));
if(!efile)
exit(0);
emp.showit();
}
}
```

Q2. Declare a structure in c++ telerec, containing name (20 characters) and telephone numbers .A binary data file "TELE.DAT" stores data of the type telerec. Write functions in c++ to do the following:

To append records in the file Display the name foe a given telephone number. If the telephone number does not exist then display error message: "Record not found".

Answer:-

(a)

```
//Function to append the record in file
void append()
{
ofstream  tfile;
tfile.open("tele.dat", ios :: app);
int  n, i;
cout << "Enter how many customers : ";
cin >> n;
for (i =0; i<n; i++)
{
cout << "\n Enter the telephone number : ";
cin >> tele.tno;
cout << "\n Enter the name : ";
gets(tele.name);
tfile.write((char *)&tele, sizeof(tele));
}
tfile.close();
}
```

(b)

```
//Function to search a record in the file
void display()
{
ifstream   tfile;
tfile.open("tele.dat", ios :: binary);
int no, flag;
flag = 0;
cout << "\n Enter telephone number to be searched : ";
cin >> no;
while(tfile)
{
tfile.read((char *)&tele , sizeof(tele));
if(!tfile)
break;
if (tele.tno == no)
{
cout << "\n Name : " << tele.name;
cout << "\n Telephone No. : " << tele.tno;
flag = 1;
}
}
if (flag == 0)
cout << "\n Record does not exist : ";
}
```

Q3. Assuming the class STOCK, write functions in c++ to perform following:-

a) Write the objects of STOCK to a binary file.

b) Read the objects of STOCK from binary file and display them on screen.

```
class STOCK
{
int ITNO;
char ITEM[10];
```

```
public:
void GETIT()
{
cin>>ITNO;
gets(ITEM);
}
void SHOWIT()
{
cout<<ITNO<<" "<<ITEM<<endl;
}
}
```

Answer:-

(a)

```
// Function to write the object of class to the binary file
void create( )
{
ofstream afile;
STOCK s;
afile.open("stock.dat", ios::out | ios :: binary);
if (!afile)
{
cout << "\n Unable to open the file ";
exit(1);
}
s.GETIT( );
afile.write((char *)&s, sizeof(s));
afile.close();
}
```

(b)

```
// Function to read the object of class from the binary file
void read_file()
{
ifstream afile;
afile.open("stock.dat", ios::in | ios :: binary);
if (!afile)
{
cout << "\n File does not exist ";
exit(1);
}
STOCK s;
afile.read((char *) &s, sizeof(s));
while(afile)
{
s.SHOWIT();
afile.read((char *) &s, sizeof(s));
}
afile.close();
}
```

Q4. Assume the class DRINKS defined below,write functions in c++ to perform the following :-

Write the objects of DRINKS to a binary file.

Read the objects of DRINKS from binary file and display them on screen when DNAME has value "INDY COLA".

```
class DRINKS
{
int DCODE ;
char DNAME [13];//Name of the drinks
int DSIZE; //size in litres
float DPRICE ;
public:
void getdrinks()
{
 cin>>DCODE>>DNAME>>DSIZE>>DPRICE;
}
void showdrinks()
{
cout<<DCODE<<DNAME<<DSIZE<<DPRICE;
}
char *getname()
{
return DNAME;
}
};
```

Answer:-

(a)

```
void read()
{
DRINKS DRI;  // Declares the class object
fstream afile; // Declare the file object
afile.open("Drink.dat", ios::app|ios::out|ios::binary); // Open the data file
int n, i;
clrscr();
cout << "Enter how many records you want to enter : ";
cin >> n;
for (i = 0; i < n; i++)
{
DRI.getdrinks(); // Call the member function to input data
afile.write((char *)&DRI, sizeof(DRINKS)); // Write the data values
}
afile.close();
}
```

(b)

```
void show()
{
```

```
DRINKS DRI;  // Declares the class object
fstream bfile;
bfile.open("Drink.dat", ios::in|ios::binary);//Open the data file
bfile.seekg(0,ios::beg);//Pointed at the 0th location in datafile
char tname[13];
if (!bfile)
cout << "File does not exist";
cout << "Enter the Name to be searched ";
gets(tname);
bfile.read((char*)&DRI,sizeof(DRINKS));//Read the data value in file
while (bfile)
{
strcpy(tname, DRI.getname());  // Enter the searched value
if (strcmp(tname, "INDYCOLA")== 0)//Compare the data value
{
cout << "\n";
DRI.showdrinks();//Display the resulted data through member function
}
bfile.read((char*)&DRI,sizeof(DRINKS));//Read the data value in file
}
bfile.close();
}
```

Q5. Consider the following class declaration:

```
class employee
{
int code;
char name[20];
float salary;
public:
void input()
{
cin>>code>>name>>salary;
}
void show()
{
cout<<code<<name<<salary<<endl;
}
float retsal()
{
return salary;
}
};
```

Give function definitions to do the following:
Write the objects of employee to a binary file.
Read the objects of employee from a binary file and display all the objects on the screen where salary is between Rs.10,000 and Rs.20,000.

Answer:

(a)

```
 void data_read()
{
employee emp;  // Declares the employee object
fstream empfile;
empfile.open("MEMP.dat", ios::app|ios::out|ios::binary);
// Creates the data file
int n, i;
clrscr();
cout << "Enter how many records you want to enter : ";
cin >> n;
for (i=0; i<n; i++)
{
emp.input();
empfile.write((char *)&emp, sizeof(employee));
}
empfile.close();
}
```

(b)

```
void data_show()
{
employee emp;  // Declares the employee object for read operation
float tsalary = 0.0;  // A temporary salary
fstream empfile;
empfile.open("MEMP.dat", ios::in|ios::binary);
empfile.seekg(0, ios::beg);
if (!empfile)
cout << "File does not exist";
empfile.read((char *)&emp, sizeof(employee)); // Reads the record one-by-one
while (empfile)
{
tsalary = emp.retsal();  // Transfer the salary into tsalary
if (tsalary >= 10000 && tsalary <= 20000)
// Checks the condition
emp.show();
// Display the output through the member function
if (empfile.eof())
// If there is no record, it terminates the loop
exit(0);
empfile.read((char *)&emp, sizeof(employee));
```

```
// Reads the record one-by-one
}
empfile.close();
}
```

Q6. Consider the following class declaration:

```
class bank
{
int accno;
char name[20];
float balance;
public:
void input()
{
cin>>accno>>name>>balance;
}
void display()
{
cout<<accno<<" "<< name<<balance<<endl;
}
float getbalance()
{
return balance;
}
};
```

Give function definitions to the following:

Write a function in c++ to accept the objects of class bank from the user and write to a binary file "BANK.DAT".

Write a function in c++ to read the objects of bank from a binary file and display all the objects on the screen where balance is more than Rs.25,000.

Answer:-

(a)

```
void data_read()
{
bank bnk;  // Declares the bank object
fstream bankfile;
bankfile.open("BANK.DAT",
ios::app|ios::out|ios::binary); // Creates the data file
int n, i;
clrscr();
cout << "Enter how many records you want to enter : ";
cin >> n;
for (i=0; i<n; i++)
{
bnk.input();
bankfile.write((char *)&bnk, sizeof(bank));
}
bankfile.close();
}
```

(b)

```
void data_show()
{
bank bnk;  // Declares the bank object for read
operation
float nbal = 0.0;;
fstream bankfile;
bankfile.open("BANK.DAT", ios::in|ios::binary);
bankfile.seekg(0, ios::beg);
if (!bankfile)
cout << "File does not exist";
bankfile.read((char *)&bnk, sizeof(bank)); // Reads the
record one-by-one
while (bankfile)
{
nbal = bnk.getbalance();
if (nbal > 250000)  // Checks the condition
bnk.display();
 // Displays the output through the member function
if (bankfile.eof())
// If there is no record, it terminates the loop
exit(0);
bankfile.read((char *)&bnk, sizeof(bank));
// Reads the record one-by-one
}
bankfile. close();
}
```

Q7. Consider the class declaration

```
class BUS
{
int bus_no;
char destination[20];
float distance;
public:
void Read();//To read an object from the keyboard
void Write();//To write an object into a file
void Show();//To display the file contents on the
monitor
};
```

Answer:

```
void BUS::Read( )
{
cout << "Enter the value of the Bus number : ";
cin >> bus_no;
cout << "Enter the value of the destination : ";
gets(destination);
cout << "Enter the value of the distance : ";
```

```
cin >> distance;
}
void BUS :: Write(BUS F)
{
fstream Ffile;
Ffile.open("Bus.dat", ios::app);
Ffile.write((char *)&F, sizeof(F));
Ffile.close();
}
void BUS::Show()
{
cout << "\n Bus No : " << bus_no;
cout << "\n Destination : " << destination;
cout << "\n Distance : " << distance;
}
```

Q8. Considered the class declaration

```
class FLIGHT
{
int flight_no;
char destination[20];
float distance;
public:
void INPUT();//To read an object from the keyboard
void Write_File(); // To write N objects into a file ,where N is passed as argument
void OUTPUT();// To display the file contents on the monitor
};
```

Complete the member functions definitions.

Answer:-

```
void FLIGHT::INPUT()
{
cout << "Enter the value of the flight number : ";
cin >> flight_no;
cout << "Enter the value of the destination : ";
gets(destination);
cout << "Enter the value of the distance : ";
cin >> distance;
}
void FLIGHT :: Write_File(FLIGHT F,int N)
{
fstream  Ffile;
Ffile.open("Flight.dat", ios::app);
for(int i=0; i<N; i++)
{
F.INPUT();
Ffile.write((char *)&F, sizeof(FLIGHT));
}
Ffile.close();
}
void FLIGHT::OUTPUT()
{
FLIGHT F;
ifstream Ffile;
Ffile.open("Flight.dat", ios::in);
Ffile.read((char *)&F, sizeof(FLIGHT));
while(Ffile)
{
if (!Ffile)
break;
cout << "\n Flight No : " << F.flight_no;
cout << "\n Destination : " << F.destination;
cout << "\n Distance : " << F.distance;
Ffile.read((char *)&F, sizeof(FLIGHT));
}
Ffile.close();
}
```

Q1. Observe the program segment given below carefully and fill in the blanks marked in statement1 and statement2 using seekp() and seekg() functions for performing the required task.

```
class Item
{
int Ino;char Item[20];
public:
//Function to search and display the content from a particular record number
void Search(int);
// Function to modify the content from a particular record number
void Modify(int);
};
void Item:: Search(int RecNo)
{
fstream File;
File.open("STOCK.DAT",ios::binary|ios::in);
_____ // statement1
file.read((char*)this,sizeof(Item));
cout<<Ino<<"□"<<Item<<endl;
File.claos();
}
void Item::Modify(int RecNo)
{
fstream File;
File.open("STOCK.DAT",ios::binary|ios::in|ios::out);
cout>>Ino;cin.getline(Item,20);
_____ ??Statment 2
File.write((char*)this,sizeof(Item));
```

```
File.close();
}
```
Answer:
```
File.seekg(RecNo*sizeof(Item));          //Statement 1
File.seekp(RecNo*sizeof(Item));          //Statement 2
```
Q2. Observe the program segment given below carefully and fill in the blank marked in Statement 1 and Statement 3 using seekg() and tellg() functions for performing the required task:
```
class Employee
{
int Eno;char ename[20];
public:
//Function to count the total number of records
int Countrec();
};
int Item::Countrec()
{
fstream File
File.open("EMP.DAT",ios::binary|ios::in);
_____ // statement 1
int Bytes=_____ //statement 2
int Count = Bytes / sizeof(Item);
File.close();
return Count;
}
```
Answer:-
```
File.seekg(0,ios::end); //statement 1
File.tellg();       //statement 2
```
Q3. Observe the program segment given below carefully, and answer the question that follows : 1
```
class Labrecord
{
int Expno;
char Experiment[20];
char Checked;
int Marks;
public :
//function to enter Experiment details
void EnterExp( );
//function to display Experiment details
void ShowExp ( ) ;
//function to return Expno
char RChecked ( ) {return Checked;
}
//function to assign Marks
void Assignmarks(int M)
{
Marks = M;
}
};
void MpdifyMarks()
{
fstream File;
File.open("Marks.Dat",ios::binary|ios::in|ios::out);
Labrecord L;
int Rec = 0;
while (File.read((char*)&L, sizeof(L)))
{
if(L.RChecked( )== ' N ' )
L.Assignmarks(0)
else
L.Assignmarks(10)
_____ //statement 1
_____ //statement 2
Rec ++ ;
}
File.close ();
}
```
If the funption ModifyMarks() is supposed to modify Marks for the records in the file MARKS.DAT based on their status of the member Checked (containing value either V or 'N'). Write C++ statements for the statement 1 and statement 2, where, statement 1 is required to position the file write pointer to an appropriate place in the file and statement 2 is to perform the write operation with the modified record.

Answer :

(a) Statement 1:
```
File.seekg(-1*sizeof(L),ios::cur);
```
OR
```
File.seekg(Rec*sizeof(L));
```
OR
```
File.seekp(-1*sizeof(L),ios::cur);
```
OR
```
File.seekp(Rec*sizeof(L));
```
OR

Any equivalent correct method of calculating size of the record in place of sizeof operator.

Statement 2:
```
File.write((char *) &L,sizeof(L));
```
OR

Any equivalent correct method of calculating size of the record in place of sizeof operator.

(½ Mark for each correct statement)

Q4. Observe the program segment given below carefully, and answer the question that follows:

```
class PracFile
{
intPracno;
char PracName[20];
int TimeTaken;
int Marks;
public:
// function to enter PracFile details
void EnterPrac( );
// function to display PracFile details
void ShowPrac( ):
// function to return TimeTaken
int RTime() {return TimeTaken;}
// function to assign Marks
void Assignmarks (int M)
{
 Marks = M;
}
};
void AllocateMarks( )
{
 fstreamFile;
File.open("MARKS.DAT",ios::binary|ios::in|ios::out);
PracFile P;
int Record = 0;
while (File.read(( char*) &P, sizeof(P)))
{
if(P.RTime()>50)
P.Assignmarks(0)
else
P.Assignmarks(10)
_____ //statement 1
_____ //statement 2
Record + + ;
}
File.close();
}
```

If the function AllocateMarks () is supposed to Allocate Marks for the records
in the file MARKS.DAT based on their value of the member TimeTaken.
Write C++ statements for the statement 1 and statement 2, where,
statement 1 is required to position the file write pointer to an appropriate
place in the file and statement 2 is to perform the write operation with the modified record.

Answer:-

(a) Statement 1:

File.seekp(Record * sizeof(P));
OR
File.seekp(Record * sizeof(PracFile));
OR
File.seekp(-sizeof(P), ios::cur);
OR
File.seekg(Record * sizeof(P));
OR
File.seekg(Record * sizeof(PracFile));
OR
File.seekg(-sizeof(P), ios::cur);
OR
Any equivalent correct method of calculating size of the record in place of sizeof operator.

Statement 2:

File.write((char*)&P, sizeof(P));
OR
File.write((char*)&P, sizeof(PracFile));
OR
Any equivalent correct method of calculating size of the record in place of
sizeof operator.

(½ Mark for each correct statement)

Q5. Observe the program segment given below carefully ,and answer the question that follows:- 1

```
class Book;
{
 int Book_no;
char Book_name[20];
public:
//function to enter Book details
void enterdetails()
//function to display Book details
void showdetails();
//function to return Book_no
int Rbook_no(){return Book_no;}
};
void Modify (Book NEW)
{
fstream File;
File.open("BOOK.DAT",ios::binary|ios::in|ios::out);
Book OB;
int Recordsread=0 ,Found=0;
while(!Found && File.read((char*)&OB,sizeof(OB)))
{
Recordsread++;
```

```
if(NEW.RBook_no()==OB.RBook_no())
{
_____ //Missing Statement
File.write((char*)&NEW,sizeof(NEW));
Found=1;
}
else
File.write((char*)&OB,sizeof(OB);
}
if (!Found)
cout<<"Record for modification does not exist";
File.close();
}
```

If the function Modify() is supposed to modify a record in file BOOK.DAT with the value of Book NEW passed to its argument ,write the appropriate statement for missing statemet using seekp() or seekg() ,whichever needed ,in the above code that would write the modicfied record at its proper place.

Answer:-

```
File.seekg(-1*sizeof(NEW),ios::out);
```

Q6. S "student.dat" file exist,with the object of class students .Assuming ,the file has just been opened through the object fil of fstream class Give a single command to place the file pointer to the third record from beginning.

In continuation to the above, command ,give a command to bring file pointer to the beginning of last record.

Answer:-

```
fil.seekg(3*sixeof(student),ios::beg);
fil.seekg(sizeof(student),ios::end);
```

ASSIGNMENT (DATA FILE HANDLING)

1. Write a user defined function in C++ to read the content from a text file STORY.TXT, count and display the number of alphabets present in it.
2. Write a user defined function in C++ to read the content from a text file NOTES. TXT, count and display the number of blank spaces present in it.
3. Assuming the class EMPLOYEE given below, write functions in C++ to perform the following:-

(i) Write the objects of EMPLOYEE to binary file.

(ii) Reads the objects of EMPLOYEE from binary file and display them on screen.

```
Class EMPLOYEE
        {
      int ENC;
            char  ENAME[0];
            PUBLIC:
Void GETIT(){ cin>> ENO;gets(ENAME);}
Void  SHOWIT() { cout>> ENO<<ENAME;<<endl; }
        };
```

4. Assuming the class STOCK, write functions in C++ to perform following:

(i) Write the objects of STOCK to binary file.

(ii) Reads the objects of STOCK from binary file and display them on screen.

```
Class STOCK
 {
      int  ITNO;
            char ITEM[10];
PUBLIC:
Void GETIT() { CIN>> ITNO; gets(ITEM);}
Void SHOWIT() { cout<<ITNO<< "  " <<ITEM<<endl ;}
 };
```

5. Consider the class declaration

```
Class BUS
 {
      int  bus_no;
```

```
        char destination[20];
        float distance;
    PUBLIC:
        void read();
        void Write();
        void show();
};
```

Complete the member functions definitions.

6. Consider the following class declaration:

```
Class FLIGHT
{
        int flight_no;
        char destination[20];
        float distance;
  PUBLIC:
 void INPUT();
      void Write_File();
        void OUTPUT();
};
```

Compute the member functions definitions.

7. Write a C++ program, which reads one line at a time form the disk file TEST . TXT , and displays it to a monitor. Your program has to read all the contents of the file. Assume the length of the line not to exceed 80 characters. You have to include all the header files if required.
8. Assuming a binary file JOKES.DAT is containing objects belonging to a class JOKE(as defined below). Write a user defined function in C++ to add more objects belonging to class JOKE at the bottom of it.

```
Class JOKE
{
        int Jokeid;
        char Jokedesc[255];
    PUBLIC:
        void Newjokeentry()
        { cin>> Jokeid; gets(type);
gets(Jokedesc); }
        void showjoke()
        { cout<<Jokeid<<"
"<<Type<<endl<<Jokedesc<<endl; }
    };
```

9. Asuming a binary file FUN.DAT is containing objects belonging to a class LAUGHTER(as defined below). Write a user defined function in C++ to add more objects belonging to class LAUGHTER at the bottom of it.

```
Class LAUTER
{
        int idno;
        char Type[5];
        char Desc[255];
    PUBLIC:
 void Newentry(){
  cin>> Idno; gets(Type); gets(Desc);}
 void Showonscreen()  {
cout<<Idno<<" "<<Type<<endll<<Desc<<endl;}
};
```

10. Assuming that a text file named TEXT1.TEXT already contains some text written into it, write a function named vowelwords(), that reads the file TEXT1.TEXT and creates a new file named TEXT2.TEXT, which shall contain only those words from the file TEXT1.TEXT which does not start with an uppercase vowel(i.e, with 'A','E','I','O','U'). FOR example, if the file TEXT1.TXT contains.

Carry umbrella and Overcoat when it Rains

then the file TEXT2.TXT shall contain

Carry when it Rains.

11. Differentiate between read and get function of istream class.
12. Write a C++ program, which initializes a string variable to the content. " Anil is a great teacher but unfortunately it kills all its pupils. Harimohan" and output the srtring one character at a time to the disk file OUT.TXT. You have to include all the header files if required.
13. A text file named "report.txt" exists on a disk. Write a program to create its copy named "Finerep.txt", which should be in small letters but the first letter of the file and first alphabetic character following a full stop should be in uppercase.
14.. Write a function in C++ to search for BookNo from a binary file "BOOK.DAT", assuming the binary file is contained the objects of the following class:

```
class BOOK
{
        int Bno; char Title[20];
        PUBLIC:
         int Rbno(){return Bno;}
         void
Enter(){cin>>Bno;gets(Title);}
        void
Display(){cout<<Bno<<Title<<endl;}
};
```

ANSWERS DATA FILE HANDLING

```
1.void display()
{
ifstream afile;
afile.open("STORY.TXT");
char ch;
int c=0;
while(afile)
{
afile.get(ch);
if(isalpha(ch))
```

```
c++;
}
cout<<"The number of alphabets are"<<c;
}

2. //COUNT AND DISPLAY THE BLANK SPACES
void display()
{
ifstream afile;
afile.open("NOTES.TXT");
char ch;
int c=0;
while(afile)
{
afile.get(ch);
if(ch= =' ')
c++;
}
cout<<"The number of blank spaces are"<<c;

3 (i)//function  to write the object of class to
 // Binary file
viod create(EMPLOYEE  emp)
{
ofstream afile;
afile.open("Emp.dat",ios::out|ios::binary);
if(!afile)
{
cout<<"\n Unable to open the file";
exit(1);
}
emp.GETIT();
afile.write((char *)&emp,sizeof(emp));
afile.close();
}
(ii) Function to read the object of clas from binary file
 void read_file()
{
ifstream afile;
afile.open("Emp.dat",ios::in|ios::binary):
if(!afile)
{
cout<<"\n File does not exist";
exit(1);
}
EMPLOYEE emp;
While(afile)
{
afile.read((char*) &emp,sizeof(emp));
emp.SHOWIT();
}
afile.close();
}

5 . //Function to write the objects
 void BUS::Write(BUS F)
{
fstream ffile;
Ffile.open("Bus.dat",ios::app);
Ffile.write((char*&F,sizeof(BUS));
Ffile.close();

//General function to read and show the objects
void BUS::Read()
{
cout<<"Enter the value of the filght number";
cin>>bus_no;
cout<<"Enter the value of the destination";
gets(destination);
cout<<"enter the value of the distance";
cin>>distance;
}
void BUS::Show()
{
cout<<"\n Flight No:"<<bus_no;
cout<<"\n Destination:"<<destination;
cout<<"\n Distance:"<<distance;
}
6. The complete member functions are as:
void FLIGHT::INPUT()
{
cout<<"Enter the value of the flight number";
cin>>flight _no;
cout<<"Enter the value of the destination";
gets(destination);
cout<<"Enter the value of the distance";
cin>>distance;
}
void  FLIGHT::WRITE_FILE(int n, FLIGHT F)
{
fstream Ffile;
Ffile.open("Flight.dat",ios::app);
For(int i=0;i<n;i++)
{
F.input();
Ffile.write((char*)&F,sizeof(FLIGHT));
}
Ffile.close();
}
void FLIGHT::OUTPUT(FLIGHT F)
{
ifstream ffile;
Ffile.open("Flight.dat");
While(Ffile)
{
Ffile.read((char*)&F,sizeof(FLIGHT));
If(!ffile)
break;
cout<<"\n Flight No:"<<F.flight_no;
cout<<"\n  Destination:"<<F.destination;
cout<<"\n Distance:"<<F.distance:
}
```

FILE HANDLING set 2

Q1) Assuming the class Computer as follows :

```
class computer
{charchiptype[10];
int speed;
public:
voidgetdetails()
{gets(chiptype);
cin>>speed;}
voidshowdetails()
{cout<<"Chip"<<chiptype<<" Speed= "<<speed;}};
```

Write a function **readfile( )** to read all the records present in an already existingbinary file SHIP.DAT and display them on the screen, also count the number ofrecords present in the file.

Q2) Consider the class declaration:

```
class BUS
{ intbus_no;
char destination[20];
float distance;
public :
void Read(); // To read an object from the keyboard
void Write (); // To write an object into a file
void Show (); // To display the file contents on The monitor };
```

Complete the member functions definitions.

Q3) Observe the program segment given below carefully and fill the blanks marked as Statement 1 and Statement 2 using seekg( ), seekp( ), tellp( ) and tellg() functions for performing the required task. **1**

```
# include <fstream.h>
class PRODUCT
{intPno; char pname [20]; intqty;
public :
:
voidModifyQty ( ); // The function is to modify quantity of a
PRODUCT};
void PRODUCT :: ModifyQty ( )
{fstreamFil;
Fil.open ("PRODUCT.DAT" , ios : : binary | ios : : in | ios : : out);
intMPno;
cout<<"Product No to modify quantity :"; cin>>MPno;
while(Fil.read ((char*) this, sizeof (PRODUCT)))
{if (MPno ==Pno)
{
cout<<"present quantity:" <<Qty<<endl;
cout<<"changed quantity:"; cin>>Qty;
int position = _____; //Statement 1
_____; //Statement 2
Fil.write ((char*) this, sizeof (PRODUCT)); // Re-writing the Record
}
}
Fil.close( );
}
```

Basic Programs

WAP to read character from file and :

4. Character.
5. Upper case character
6. Lower case character
7. Digits
8. Other characters/Symbol
9. vowel character
10. consonant character
11. Spaces
12. Words
13. Lines
14. Character from user

WAF to read Words from file and

15. Words.
16. Upper case character
17. Lower case character
18. Digits character
19. Special character
20. Words end with Upper case character
21. Words end with Lower case character
22. Words end with Digits character
23. Words end with Special character
24. and end with vowel character
25. and end with consonant character
26. vowel and end with consonant character
27. consonant and end with vowel character
28. Vowel and consonant present in each word .
29. Search/ and count a given word dynamic
30. Search/and count a given word Static like is, am, are, this, C++

WAF to read Line from file and

31. lines.
32. lines.

each line.

33. start with Upper case character
34. start with Lower case character
35. start with Digits character
36. start with Special character
37. end with Upper case character
38. end with Lower case character
39. end with Digits character
40. end with Special character
41. start with and end with vowel character
42. start with and end with consonant character
43. start with vowel and end with consonant character
44. start with consonant and end with vowel character
45. vowel and consonant present in each word in line.

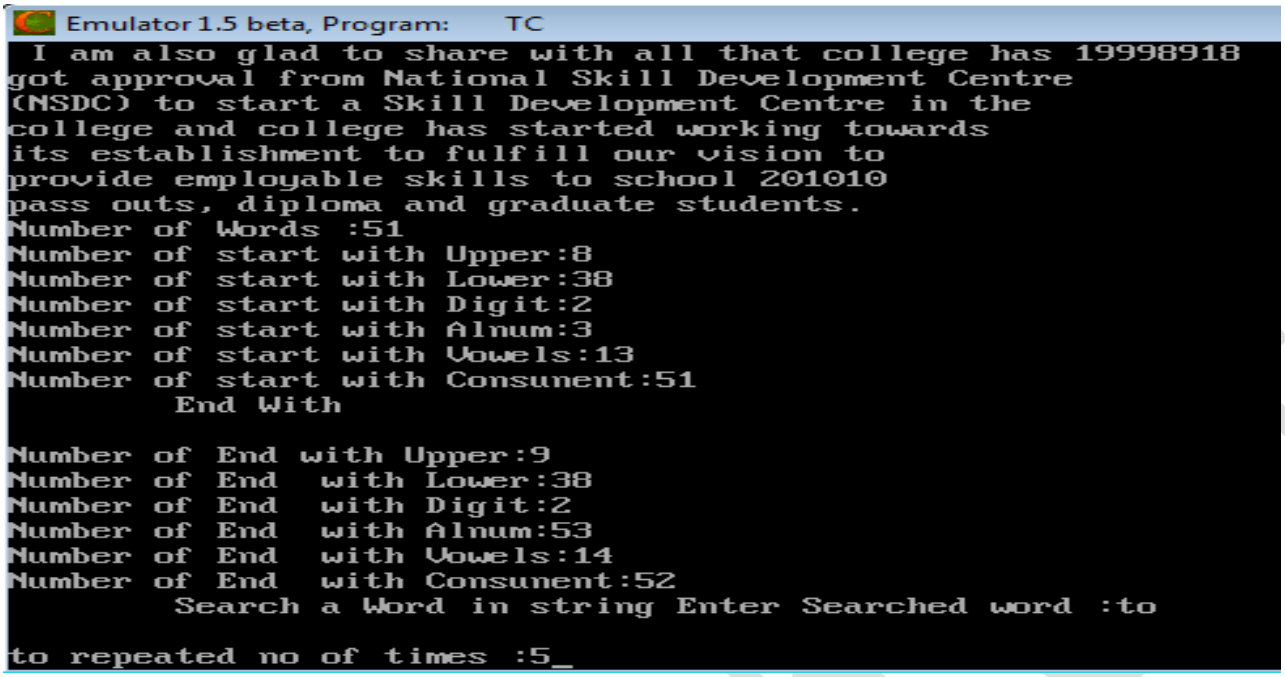
P1. Character

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<ctype.h>
#include<dos.h>
void main()
{
char msg[30];
clrscr();
int s=0,w=0,l=0,u=0,d=0,sc=0,co=0,v=0,a=0,an=0,n=0;
ifstream y;
y.open("x.txt",ios::in);
int c=0;
char ch;
while(y)
{
y.get(ch);
delay(20);
if(isspace(ch))
s++;
if(isupper(ch))
u++;
if(islower(ch))
l++;
if(isupper(ch))
u++;
if(isdigit(ch))
d++;
if(isalpha(ch))
a++;
if(isalnum(ch))
an++;
if(!isalnum(ch))
sc++;
if(ch=='\n')
n++;
c++;
cout<<ch;
}
cout<<"\nNumber of Character :"<<c;
cout<<"\nNumber of Upper Case Character :"<<u;
cout<<"\nNumber of Lower Case Character :"<<l;
cout<<"\nNumber of Special Case Character :"<<sc;
cout<<"\nNumber of Digit Case Character :"<<d;
cout<<"\nNumber of Space Case Character :"<<s;
cout<<"\nNumber of Words Character :"<<(s+1);
cout<<"\nNumber of Alpha Case Character :"<<a;
cout<<"\nNumber of AlphaNum Case Character :"<<an;
cout<<"\nNumber of lines :"<<n;
y.close();
getch();
}
```

```
I am also glad to share with all that college has 19998918
got approval from National Skill Development Centre
(NSDC) to start a Skill Development Centre in the
college and college has started working towards
its establishment to fulfill our vision to
provide employable skills to school 201010
pass outs, diploma and graduate students.
Number of Character :340
Number of Upper Case Character :24
Number of Lower Case Character :253
Number of Special Case Character :61
Number of Digit Case Character :14
Number of Space Case Character :56
Number of Words Character :57
Number of Alpha Case Character :265
Number of AlphaNum Case Character :279
Number of lines :6
```

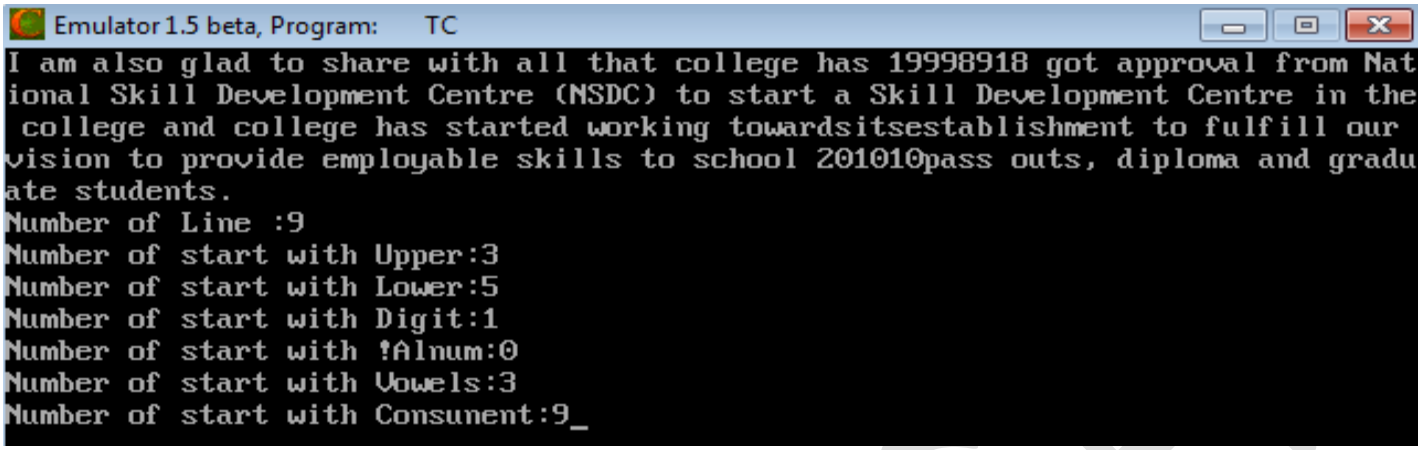
P1. Word

```
#include<iostream.h>
#include<string.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<ctype.h>
#include<dos.h>
void main()
{
char w[30];
clrscr();
int s=0,l=0,u=0,d=0,sc=0,co=0,v=0,a=0,an=0,n=0,c1=0,c=0;
ifstream y;
y.open("x.txt",ios::in);
cout<<"\n\tStart With\n";
while(!y.eof())
{
y.getline(w,20,' ');
delay(50);
if(isupper(w[0]))u++;
if(islower(w[0]))l++;
if(isdigit(w[0]))d++;
if(!isalnum(w[0]))an++;
if(w[0]=='a'||w[0]=='e'||w[0]=='i'||w[0]=='o'||w[0]=='u'||w[0
]=='A'||w[0]=='E'||w[0]=='I'||w[0]=='O'||w[0]=='U')v++;
if(isalpha(w[0]))
{
if(w[0]!='a'||w[0]!='e'||w[0]!='i'||w[0]!='o'||w[0]!='u'||w[0]!='
A'||w[0]!='E'||w[0]!='I'||w[0]!='O'||w[0]!='U')
c1++;
}
c++;
cout<<" "<<w;
}
cout<<"\nNumber of Words :"<<c;
cout<<"\nNumber of start with Upper:"<<u;
cout<<"\nNumber of start with Lower:"<<l;
cout<<"\nNumber of start with Digit:"<<d;
cout<<"\nNumber of start with Alnum:"<<an;
cout<<"\nNumber of start with Vowels:"<<v;
cout<<"\nNumber of start with Consunent:"<<c;
y.close();
y.open("x.txt",ios::in);
cout<<"\n\tEnd With\n";
while(!y.eof())
{
y.getline(w,20,' ');
//delay(50);
int l=strlen(w);
if(isupper(w[l-1]))u++;
if(islower(w[l-1]))l++;
if(isdigit(w[l-1]))d++;
if(!isalnum(w[l-1]))an++;
if(w[l-1]=='a'||w[l-1]=='e'||w[l-1]=='i'||w[l-1]=='o'||w[l-
1]=='u'||w[l-1]=='A'||w[l-1]=='E'||w[l-1]=='I'||w[l-
1]=='O'||w[l-1]=='U')v++;
if(isalpha(w[l-1]))
{
if(w[l-1]!='a'||w[l-1]!='e'||w[l-1]!='i'||w[l-1]!='o'||w[l-
1]!='u'||w[l-1]!='A'||w[l-1]!='E'||w[l-1]!='I'||w[l-1]!='O'||w[l-
1]!='U')
c++;
}
}
cout<<"\nNumber of End with Upper:"<<u;
cout<<"\nNumber of End  with Lower:"<<l;
cout<<"\nNumber of End  with Digit:"<<d;
cout<<"\nNumber of End  with Alnum:"<<an;
cout<<"\nNumber of End  with Vowels:"<<v;
cout<<"\nNumber of End  with Consunent:"<<c;
y.close();
y.open("x.txt",ios::in);
char sw[20];
cout<<"\n\tSearch a Word in string ";
cout <<"Enter Searched word :";
gets(sw);
int c2=0;
while(!y.eof())
{
y.getline(w,20,' ');
if(strcmpi(w,sw)==0)
  c2++;
}
cout<<"\n"<<sw<<" repeated no of times :"<<c2;
y.close();
getch();}
```



P3. Line

```
#include<iostream.h>
#include<string.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<ctype.h>
#include<dos.h>
void main()
{
char Line[80];
clrscr();
int s=0,l=0,u=0,d=0,sc=0,co=0,v=0,a=0,an=0,n=0,c1=0,c=0;
ifstream y;
y.open("x.txt",ios::in);
while(!y.eof())
{
y.getline(Line,80,'\n');
delay(50);
if(isupper(Line[0]))u++;
if(islower(Line[0]))l++;
if(isdigit(Line[0]))d++;
if(!isalnum(Line[0]))an++;
if(Line[0]=='a'||Line[0]=='e'||Line[0]=='i'||Line[0]=='o'||Line[0]
=='u'||Line[0]=='A'||Line[0]=='E'||Line[0]=='I'||Line[0]=='O'||L
ine[0]=='U')v++;
if(Line[0]!='a'||Line[0]!='e'||Line[0]!='i'||Line[0]!='o'||Line[0]!='
u'||Line[0]!='A'||Line[0]!='E'||Line[0]!='I'||Line[0]!='O'||Line[0]!
='U')
c1++;
c++;
cout<<Line;
}
cout<<"\nNumber of Line :"<<c;
cout<<"\nNumber of start with Upper:"<<u;
cout<<"\nNumber of start with Lower:"<<l;
cout<<"\nNumber of start with Digit:"<<d;
cout<<"\nNumber of start with !Alnum:"<<an;
cout<<"\nNumber of start with Vowels:"<<v;
cout<<"\nNumber of start with Consunent:"<<c;
y.close();
getch();
}
```



P4. Line

```
#include<iostream.h>
#include<string.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<ctype.h>
#include<dos.h>
void main()
{
char Line[80];
clrscr();
int
s=0,l=0,u=0,d=0,sc=0,co=0,v=0,a=0,an=0,n=0,c1=0,c=0;
ifstream y;
y.open("x.txt",ios::in);
while(!y.eof())
{
y.getline(Line,80,'\n');
delay(50);
int l=strlen(Line);
if(isupper(Line[l-1]))u++;
if(islower(Line[l-1]))l++;
if(isdigit(Line[l-1]))d++;
if(!isalnum(Line[l-1]))an++;
if(Line[l-1]=='a'||Line[l-1]=='e'||Line[l-1]=='i'||Line[l-
1]=='o'||Line[l-1]=='u'||Line[l-1]=='A'||Line[l-
1]=='E'||Line[l-1]=='I'||Line[l-1]=='O'||Line[l-
1]=='U')v++;
if(Line[l-1]!='a'||Line[l-1]!='e'||Line[l-1]!='i'||Line[l-
1]!='o'||Line[l-1]!='u'||Line[l-1]!='A'||Line[l-1]!='E'||Line[l-
1]!='I'||Line[l-1]!='O'||Line[l-1]!='U')
c1++;
c++;
cout<<Line;
}
cout<<"\nNumber of Line :"<<c;
cout<<"\nNumber of End with Upper:"<<u;
cout<<"\nNumber of End with Lower:"<<l;
cout<<"\nNumber of End with Digit:"<<d;
cout<<"\nNumber of End with !Alnum:"<<an;
cout<<"\nNumber of End with Vowels:"<<v;
cout<<"\nNumber of End with Consunent:"<<c;
y.close();
getch();
}
```