CHAPTER 6
DATA FILE HANDLING

SYLLABUS

- **Data File Handling**: Need for a data file, Types of data files – Text file and Binary file;
- **Text File**: Basic file operations on text file: Creating/Writing text into file, Reading and Manipulation of text from an already existing text file (accessing sequentially);
- **Binary File**: Creation of file, Writing data into file, Searching for required data from file, Appending data to a file, Insertion of data in sorted file, Deletion of data from file, Modification of data in a file; Implementation of above mentioned data file handling in C++; Components of C++ to be used with file handling: Header file: fstream.h; ifstream, ofstream, fstream classes;
  - Opening a text file in, out, and app modes; Using cascading operators (>> <<) for writing text to the file and reading text from the file; open(), get(), read(), put(), write(), getline() and close() functions; Detecting end-of-file (with or without using eof() function); tellg(), tellp(), seekg(), seekp();

**TOPIC-1**
Data File Handling: text file

Quick Review

1. **File**
   A stream of bytes stored in secondary storage devices. It is classified into two types: (i) Text file (ii) Binary file. Different streams are used to represent different types of data flow.

2. **File I/O Classes**
   The following file I/O Classes used for file read/write operations
   - ifstream (for read operations)
   - ofstream (for write operations)
   - fstream (for both read & write operations)

3. **Text file functions**
   - char I/O: different char I/O functions are used as given below:
     - get(): read a single character from text file
     - put(): writing a single character in text file
     - get line(): read a line of text from text file store in a buffer.

4. **Using Stream I/O Classes**

![Fig. 1 Using the cin object for the input stream](image-url)
5. Reading from and Writing to Files Using the I/O Classes

To keep data in permanent storage, you can input and output information to files instead of using the keyboard and the display. Streams can connect the keyboard and monitor to our program. Additionally, an input stream connect a file to our program and an output stream, connect our program to a file. These concepts are illustrated in the following diagrams:

Fig. 2 Using the cout object for the output stream

Fig. 3 Using a file for the input stream

Fig. 4 Using a file for the output stream

Fig. 5 Inheritance diagram for stream I/O classes of C++

6. Using file I/O

Files

Program Files

Data Files

Text Files

Binary Files
The language like C/C++ treat everything as a file.
The basic operation on text/binary files are: Reading/writing, reading and manipulation of data stored on these files. Both type of files needs to be open and close.

**How to open a file**:

<table>
<thead>
<tr>
<th>Using member function open ()</th>
<th>Using Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>Syntax</td>
</tr>
<tr>
<td>Filestream object;</td>
<td>Filestream object;</td>
</tr>
<tr>
<td>object.open</td>
<td>(&quot;filename&quot;, mode);</td>
</tr>
<tr>
<td>(&quot;filename&quot;, mode);</td>
<td>Example</td>
</tr>
<tr>
<td>Example</td>
<td>Example</td>
</tr>
<tr>
<td>ifstream fin;</td>
<td>ifstream fin(&quot;abc.txt&quot;);</td>
</tr>
<tr>
<td>fin.open (&quot;abc.txt&quot;);</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
(a) Mode are optional and given at the end.
(b) Filename must follow the convention of 8.3 and its extension can be anyone.

**How to close a file:**
All types of files can be closed using close () member function.

**Syntax**
```cpp
fileobject.close ();
```

**Example**
```cpp
fin.close ();
/*here fin is an object of ifstream class*/
```

<table>
<thead>
<tr>
<th>Program</th>
<th>ABC.Txt file contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>#include &lt;fstream.h&gt;</td>
<td>This is my first program in file handling</td>
</tr>
<tr>
<td>//using namespace std;</td>
<td>Hello again</td>
</tr>
<tr>
<td>void main()</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>ofstream fout;</td>
<td></td>
</tr>
<tr>
<td>fout.open (&quot;abc.txt&quot;);</td>
<td></td>
</tr>
<tr>
<td>fout&lt;&lt;&quot;This is my first program in file handling&quot;;</td>
<td></td>
</tr>
<tr>
<td>fout&lt;&lt;&quot;\n Hello again&quot;;</td>
<td></td>
</tr>
<tr>
<td>fout.close ();</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

7. Opening a file

**Opening File Using Constructor**

- ofstream outFile("sample.txt"); //output only
- ifstream inFile("sample.txt"); //input only

**Opening file using open()**

- Stream-object.open("filename", mode)
- ofstream outFile;
- outFile.open("sample.txt");
- ifstream inFile;
- inFile.open("sample.txt");

**Closing File**

- outFile.close();
- inFile.close();

**Know the Terms**

- **Binary file**: It is a file that contains information in the same format as it is held in memory. In binary files, no delimiters are used for a line and no translations occur here.
- **file**: A sequence of bytes or words holding information in a computer. The term “file” is usually reserved to information placed on disk or elsewhere outside the main memory. The file part of the C++ standard library provides ifstream, ofstream, and fstream as abstraction for accessing files.
- **file stream**: Stream attached to a file.
- **ifstream**: An file stream for input.
- **iostream**: Standard library flexible, extensible, type-safe input and output framework. Stream that can be used for both input and output.
- **istream**: Input stream type.
- **ostream**: Output stream type.
- **Stream**: It refers to a sequence of bytes.
- **Text file**: It is a file that stores information in ASCII characters. In text files, each line of text is terminated with a special character known as EOL (End of Line) character or delimiter character. When this EOL character is read or written, certain internal translations take place.

**Short Answer Type Questions-1**

Q. 1. Write a user defined function word_count() in C++ to count how many words are present in a text file named "opinion.txt". [CBSE SQP 2016]
   For Example, if the file opinion.txt contains following text:
   
   Co-education system is necessary for a balanced society. With co-education system. Girls and Boys may develop a feeling of mutual respect towards each other.
   
   The function should display the following:
   Total number of words present in the text file are: 24

   **Ans.**
   ```cpp
   void word_count()
   {
       ifstream in; char ch[20]; int c=0;
       in.open("opinion.txt");
       while(!in.eof())
       {
           ch>>in;
           c=c+1;
       }
       cout<<"Total number of words present in the text file are: "<"<c;
   }
   ```

   (½ Mark for opening opinion.txt correctly)
   (½ Mark for fetching each word from the file correctly)
   (½ Mark for counting each word)
   (½ Mark for correct display)
   [CBSE Marking Scheme 2016]

Q. 2. Write a function in C++ to count and display the no of three letter words in the file “VOWEL.TXT”. [CBSE SQP 2015]
   
   Example:
   If the file contains:
   A boy is playing there. I love to eat pizza. A plane is in the sky.
   Then the output should be: 4

   **Ans.**
   ```cpp
   #include<iostream.h>
   #include<string.h>
   void wordcount()
   {
       char word[80];
       int cnt=0;
       ifstream f1;
       f1.open("VOWEL.TXT");
       while (f1>>word)
       {
           if(strlen(word) == 3)
           {
               cnt++;
               cout<<word<<endl;
           }
       }
       cout<<" number of three letter words = "<"<cnt;
       f1.close();
   }
   ```

   [2]

Q. 3. Write a function AECount() in C++, which should read each character of a text file NOTES.TXT, should count and display the occurrence of alphabets A and E (including small cases a and e too).
   
   Example:
   If the file content is as follow:
   CBSE enhanced its CCE guidelines further.
   The AECount() function should display the output as
   A:1
   E:7
   [O.D. 2014]
   
   **Ans.**
   ```cpp
   void AECount()
   {
       Fstream obj;
       obj. open ("NOTES.TXT", ios :: in);
       char x;
       int i=0, Sum A=0, Sum E=0,
       while (obj.get (ch)!=0)
   ```
Q. 4. Write a function Countaroma() to count and display the number of times "Aroma" occurs in a file "Cook.txt". [CBSE Comp. 2014]

**Note**: Only complete word "Aroma" should be counted. Words like "Aromatic" should not be counted.

**Ans.**

```cpp
void Countaroma()
{
    ifstream tfile;
    tfile.open("Cook.txt", ios::in);
    char arr[80];
    char ch;
    int i=0; sum=0; n=0;
    while(tfile)
    {
        tfile.get(ch);
        arr[i] = ch;
        i++;
        if (strcmp(ch, "Aroma") == 0)
        {
            i--;
            sum = sum + i;
            n++;
        }
    }
    cout<<"Total number of Aroma:" <<n;
}(2 marks for correct program)
```

Q. 5. Write a function EUCount() in C++, which should read each character of a text file IMP.TXT, should count and display the occurrence of alphabets E and U (including small cases e and u too).

**Example**: If the file content is as follows:
Updated information is simplified by official websites.
The EUCount() function should display the output as
E:4
U:1

[Delhi, 2014]

**Ans.**

```cpp
void EUCount()
{
    ifstream obj;
    obj.open("IMP.TXT");
    char ch;
    int i=0, sumE=0, sumU=0;
    while (obj.get(ch) != EOF)
    {
        if (ch == 'E' || ch == 'e')
        {
            sumE = sumE + 1;
            if (ch == 'U' || ch == 'u')
            {
                sumU = sumU + 1;
            }
        }
    }
    cout << "E:" << sumE << endl;
    cout << "U:" << sumU << endl;
}(1)
```

Q. 6. Write a function CountDigit() in C++ which reads the content of a text file story.txt and displays the number of digits in it.

**For example if the file contains**:
Amrapali was a queen of Gareware Kingdom in the year 1911. She had 2 daughters.
Her palace had 200 rooms.

**Then the output on the screen should be**:
Number of digits in Story: 8 [CBSE Comp. 2013]

**Ans.**

```cpp
void CountDigit()
{
    ifstream fin("STORY.TXT");
    char ch;
    int count = 0;
    while (!fin.eof())
    {
        fin >> ch;
        if (isdigit(ch))
        {
            count++;
        }
    }
    cout << "Number of digit in Story:" << count;
    fin.close();
}(1)
```

Q. 7. Write a function in C++ to count the number of lines starting with a digit in a text file "Diary.Txt". [CBSE SQP 2013]

**Ans.**

```cpp
int countNum()
{
    ifstream fin("Diary.Txt");
    char ch[80];
    int count = 0;
    while (!fin.eof())
    {
        fin.getline(ch, 80);
        if (isdigit(ch[0]))
        {
            count++;
        }
    }
    fin.close();
    return count;
}(2)
```
Q. 8. Write a function in C++ to read the content of a text file “DELHI.TXT” and display all those lines on the screen, which are either starting with ‘D’ or starting with ‘M’.

**Answ.**

```cpp
void disp()
{
    ifstream FILE("DELHI.TXT");
    int CA=0;
    char LINE [80];
    while(FILE.getline(LINE,80))
        if(LINE[0] == 'D' || LINE[0] == 'M')
            puts(LINE);
    FILE.close();
}
```

**[O.D, 2012]**

Q. 9. Write a function in C++ to read the content of a text file “PLACES.TXT” and display all those lines on screen, which are either starting with ‘P’ or starting with ‘S’.

**Answ.**

```cpp
void disp()
{
    ifstream FILE("PLACES.TXT");
    int CA=0;
    char LINE [80];
    while(FILE.getline(LINE,80))
        if(LINE[0] == 'P' || LINE[0] == 'S')
            puts(LINE);
    FILE.close();
}
```

**[Delhi, 2012]**

Q. 10. Write a function CountYouMe ( ) in C++ which reads the contents of a text file story.txt and counts the words You and Me (not case sensitive). For example, if the file contains:
    You are my best friend.
    You and me make a good team.

**Answ.**

```cpp
void CountYouMe()
{
    ifstream FILE("story.txt");
    char word [80];
    int WC = 0, WM = 0;
    while (! FILE.eof())
    {
        FILE>>word;
        if ((strcmp (word, "You") == 0))
            WC++;
        if ((strcmp (word, "Me") == 0))
            WM++;
    }
    cout <<"count for you":<< WC;
    cout <<"count for me":<< WM;
    FILE.close();
}
```

**Ans.**

```cpp
void CountLine()
{
    ifstream FILE("STORY.TXT");
    int LINES=0;
    char STR[80];
    while
    {
    FILE.getline(STR,80);
    LINES++;
    }
    cout <<"No. of Lines":<<LINES;
    FILE.close();
}
```

**Q. 11. Write a function in C++ to count the number of lines present in a text file “STORY.TXT”.

**Ans.**

**Quick Review**

1. **Binary File**
   Containing information in the non-readable form

2. **Binary File Functions**
   - read () : reads a block of binary data or reads a fixed number of bytes from the specified stream.
   - write () : writes a block of binary data.

3. **Input and Output Operation**
   - put() and get() function
The function put() writes a single character to the associated stream. While, the function get() reads a single character from the associated stream.

demo:
file.get(ch);
file.put(ch);

**write() and read() function**
Both write() and read() functions write and read blocks of binary data.

demo:
file.read((char *)&obj, sizeof(obj));

file.write((char *)&obj, sizeof(obj));

4. **Writing/Reading Data in Binary Format**
To write and read data in binary format, two member functions are available in C++. They are:
read() and write().

Fileobject.write((char *)&object, size of(object));
Fileobject.read((char *)&object, size of(object));

Example of write() member function.

```cpp
#include <fstream.h>
#include <iostream.h>

struct student
{
  int roll;
  char name[30];
  char address[60];
};

int 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, size of(student));
  fout.close();
  return 0;
}
```

**Writing class object in a file**

```cpp
#include <fstream.h>
#include <iostream.h>

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()
{
  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;
}

5. Some other important functions

<table>
<thead>
<tr>
<th>Member function name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>seekg()</td>
<td>Used to move reading pointer forward and backward</td>
</tr>
<tr>
<td>Syntax</td>
<td>fileobject.seekg(no_of_bytes, mode);</td>
</tr>
<tr>
<td>Example:</td>
<td>(a) fout.seekg(50,ios::cur);     // move 50 bytes forward from current position</td>
</tr>
<tr>
<td></td>
<td>(b) fout.seekg(50,ios::beg);      // move 50 bytes forward from current beginning</td>
</tr>
<tr>
<td></td>
<td>(c) fout.seekg(50,ios::end);      // move 50 bytes forward from end.</td>
</tr>
<tr>
<td>seekp()</td>
<td>Used to move writing pointer forward and backward</td>
</tr>
<tr>
<td>Syntax</td>
<td>fileobject.seekp(no_of_bytes, mode);</td>
</tr>
<tr>
<td>Example:</td>
<td>(a) fout.seekp(50,ios::cur);     // move 50 bytes forward from current position</td>
</tr>
<tr>
<td></td>
<td>(b) fout.seekp(50,ios::beg);      // move 50 bytes forward from current beginning</td>
</tr>
<tr>
<td></td>
<td>(c) fout.seekp(50,ios::end);      // move 50 bytes forward from end.</td>
</tr>
<tr>
<td>tellp()</td>
<td>It return the distance of writing pointer from the beginning in bytes</td>
</tr>
<tr>
<td>Syntax</td>
<td>Fileobject.tellp();</td>
</tr>
<tr>
<td>Example:</td>
<td>long n = fout.tellp ( );</td>
</tr>
<tr>
<td>tellg()</td>
<td>It returns the distance of reading pointer from the beginning in bytes</td>
</tr>
<tr>
<td>Syntax</td>
<td>Fileobject.tellg();</td>
</tr>
<tr>
<td>Example:</td>
<td>long n = fout.tellg();</td>
</tr>
</tbody>
</table>
6. File Modes

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

7. Text files

In these types of files, all the data is first converted into their equivalent char and then it is stored in the files.

8. Binary files

In these types of files, all the data is stored in the binary format as it is stored by the operating system, there is no conversion. Hence, the processing speed is much more than text files.

<table>
<thead>
<tr>
<th>get() member function</th>
<th>getline() function</th>
</tr>
</thead>
<tbody>
<tr>
<td>get() function is used to read a single char from the input stream in text file.</td>
<td>getline() function is used to read a string from the input stream in text file.</td>
</tr>
<tr>
<td>Syntax</td>
<td>Syntax</td>
</tr>
<tr>
<td>fileobj.get(char);</td>
<td>fileobj.getline(string, no_of_char, delimiter);</td>
</tr>
<tr>
<td>Example:</td>
<td>Example:</td>
</tr>
<tr>
<td>fin.get(ch);//fin is file stream.</td>
<td>fin.getline(str, 80);//fin is file stream.</td>
</tr>
<tr>
<td>Note: Delimiter is optional.</td>
<td></td>
</tr>
</tbody>
</table>

Know the Terms

- **binary operator**: An operator taking two operands, such as /, &, and binary *.
- **bit**: A unit of memory that can hold 0 or 1. An individual bit cannot be directly accessed in C++ (the unit of addressing is a byte), but a bit can be accessed through a bit field or by using the bitwise logical operators & and |.
- **bitand**: Synonym for &, the bitwise and operator.
- **bitfield**: A number of bits in a word made accessible as a struct member.
- **bitor**: Synonym for |, the bitwise or operator.
- **byte**: A unit of memory that can hold a character of the C++ representation character set. The smallest unit of memory that can be directly addressed in C++. Usually, a byte is 8 bits.

Very Short Answer Type Questions

Q. 1. Write the command to place the file pointer at the 10th and 4th record starting position using seekp() or seekg() command. File object is "file" and record name is "STUDENT". [CBSE SQP 2015]

**Ans.**

file.seekp(9 * sizeof(STUDENT), ios::beg); [½]

file.seekp(3 * sizeof(STUDENT), ios::beg); [½]

Q. 2. Find the output of the following C++ code considering that the binary file sp.dat already exists on the hard disk with 2 records in it.

```cpp
class sports {
    int id;

    void reading()
    {
        ifstream i;
        i.open("sp.dat");
    }
}
```

- char sname[20];
- char coach[20];
- public:
- void entry();
- void show();
- void writing();
- void reading();
```cpp```
while(1)
{
  i.read((char*)&s.size of (s));
  if(i.eof())
    break;
  else
    cout<<"\n"<<i.tellg();
}
i.close();
}

void main()
{
  s.reading();
}

Ans. 42

84

(½ Mark for each correct answer)

Q. 3. A binary file “games.dat” contains data of 10 games where each
game’s data is an object of the following class:

[CSET Compt. 2014]

class game
{
  int gameno;
  char game_name[20];
  public:
    void enterdetails(){cin>gameno,
    gets(game_name);}
    void enterdetails()
    {cout<gameno<endl< game_name;}
}

With reference to this information, write C++ statement in the blank
given below to move the file pointer to the end of file.

ifstream ifile;
game G;
ifile.open("games.dat",ios::binary| 
ios::in);
cout<ifile.tellg();

Ans. ifile.seekg(0,iost::end); [1]

Q. 4. Fill in the blanks marked as Statement 1 and
Statement 2, in the program segment given
below with appropriate functions for the
required task.

class Agency
{
  int ANo;       //Agent Code
  char AName [20];  //Agent Name
  char Mobile [12];  //Agent Mobile
  public:
    void Enter ();  //Function to enter details of agent*
    void Disp ();  //Function to display details of agent*/
    int RAno () {return ANo;}
    void UpdateMobile()  //Function
to update Mobile*/
    {
      cout<"Updated Mobile:";
      gets(Mobile);
    }
  void AgentUpdate ()
  {
    fstream F;
    F. open ("AGENT.DAT", ios::
    binary | ios:: in | ios:: out);
    int Updt=0;
    int UAno;
    cout<<"Ano (Agent No - to update
    Mobile):";
    cin>> UAno;
    Agency A;
    while ((!Updt && F.read ((char*) &A,
    sizeof (A))))
    {
      if (A. RAno ()==UAno)
      {
        //Statement 1: To call the function
to Update Mobile No.
        //Statement 2: To reposition file
        pointer to re-write the updated
        object back in the file
        F. Write ((char*) &A, sizeof (A));
        Updt++;
        }
        if (Updt)
        cout<<"Mobile Updated for Agent"
        <<UAno<<endl;
        else
        cout<<"Agent not in the
        Agency"<<endl;
        F. close ();
      }
    [O.D. 2014]
    Ans. Statement 1:
    A. Update Mobile ();
    Statement 2:
    F. seekg(-1*size of (A),ios::cur); [1]

Q. 5. Fill in the blanks marked as Statement 1 and
Statements 2, in the program segment given
below with appropriate functions for the
required task.  

[Delhi, 2014]

class Medical
{
int RNo;  //Representative Code
char Name[20];  
    //Representative Name
char Mobile[12];  
    //Representative Mobile

public:
    void Input();  // Function to enter all details
    void Show();  //Function to display all details
    int RRno (){return RNo;}
    void Change Mobile () //Function to change Mobile
    {
        cout<"Changed Mobile:";
        gets (mobile);
    }
};

void RepUpdate ()
{

    fstream F;
    F.open ("REP. DAT", ios: :binary|ios::in|ios::out):
    int Change=0;
    int URNo;
    cout<"Rno (Rep No-to update Mobile) :");
    cin>>URNo;
    Medical M;
    while ( (!Change && F.read (char*) &M, sizeof (M))))
    {
        if (M.RRRn)==URNo)
        {
            //Statement 1: To call the function to change Mobile No.
            --------------------
            //Statement 2: To reposition file pointer to re-write
            //the updated object back in the file  
            F.write ((char*) &M, sizeof (M));
            Change++;
        }
    }
    if (change)
        cout <"Mobile Changed for Rep"
        <<URNo<<endl;
    else
        cout<"Rep not in the Medical"<<end1;
    F.close ();
}

Ans. Statement 1:
M. Change Mobile ();
Statement 2:
F. seekp (-1*sizeof (M), ios::ur);

Q. 6. Fill in the blanks marked as Statement 1 and Statement 2, in the program segment given below with appropriate functions for the required task.
class Agent{
    long ACode;  //Agent Code
    char AName[20];  //Agent Name
    int Commission;

public:
    void Enter ();  //Function to enter details of Agent
    void Display ();  //Function to display details of Agent
    void Update (int c)  //Function to modify commission
    {
        Commission = C;
    }
    int GetComm(){return Commission;}
    long GetAcode() {return Acode;}
};
void ChangeCommission (long AC, int CM)
    //AC “ Agent Code, whose commission needs to change
    //CM “ New Commission
    
    fstream F;
    F.open ("AGENT.DAT", ios ::binary |ios ::in|ios ::out);
    char Changed='N';
    Agent A;
    while (Changed==’N’ && F.read (char*) &A, sizeof (A)))
    {
        if (A.GetAcode ()==AC)
        {
            Changed = ‘Y’;
            A. Update (CM);
            //Statement 1: To place file pointer to the required position
            //Statement 2: To write the object A on to the binary file
        }
    }
    if (Changed==’N’)
        cout<"Agent not registered . . .";
    F.close ();  

[CBSE Comptt., 2013]
Ans. Statement 1:
F. seekg(position);
Statement 2:
F. write((char*) &A, sizeof(A));  

Q. 7. Fill in the blanks marked as Statement 1 and Statement 2, in the program segment given below with appropriate functions for the required task.

```cpp
class Customer
{
    long int CNo;  //Customer Number
    char CName[20];  //Customer Name
    char Email[30];  //Email of Customer
    public:
        void Allocate();  //Function to allocate a member
        void Show();  //Function to show customer data
        void ModifyEmail()  //Function to modify Email
            {cout<<"Enter Modified Email:";
            gets(Email);
            }
    long int GetCno() {return CNo;}
};
void ChangeData()
{
    fstream File;
    File.open("CUST.DAT", ios::binary | ios::in | ios::out);
    int Change=0, Location;
    long int ChangeCno;
    cout<<"Cno - whose email required to be modified:";
    cin>>ChangeCno;
    Customer CU;
    while(Modify && File.read((char*) &CU, sizeof(CU))
    {if (CU.GetCno() == ChangeCno)
        {CU.ModifyEmail();
        Location=File.tellg() - sizeof(CU);
        //Statement 1: To place file pointer to the required position
        __________;
        //Statement 2: To write the object CU on to the binary file
        __________;
        Change ++;
        }
    }
```  

Ans. Statement 1: File.seekp(-1*sizeof(CU), ios::cur);
Statement 2: File.write((char*) &CU, sizeof(CU));

Q. 8. A binary file "Students.dat" contains data of 10 students where each student's data is an object of the following class:

```cpp
class Student
{
    int Rno; char Name[20];
    public:
        void EnterData() {cin>>Rno; cin.getline(Name, 20)}
        void showData() {cout<<Rno<<"-"
            <<Name<<endl;}
};

With reference to this information, write output of the following program segment:
```  

```cpp
ifstream file; Student S;
File.open("Students.dat", ios::binary | ios::in);
File.seekg(0, ios::end);
Cout<<File.tellg();  
```  

Ans. A1
B2
C3

Q. 9. Observe the program segment given below carefully and answer the questions that follow.

```cpp
class Stock
{
    int Ino, Qty; char Item[20];
    public:
        void Enter() {cin>>Ino;
            gets(Item); cin>>Qty;
            void Issue (int Q) {Qty += Q;}
            void Purchase (int Q) {Qty -= Q;}
            int GetIno() {return Ino;}
        }
        void PurchaseItem (int Pno, int PQty)
        {fstream File;
            File.open("STOCK.DAT", ios::binary | ios::in | ios::out);
            Stock S;
            int Success=0;
```  

```cpp
```
while (Success==0 && File.read((char*) &S, sizeof (S)))
{
    if (Pino==S.GetIno ())
    {
        S. Purchase (PQty);
        //Statement 1
        //Statement 2
        Success++;
    }
}

} if (Success==1)
    cout<<" Purchase Updated" <<endl;
else
    cout<<"Wrong Item No"<<endl;
File . close ( );

(i) Write statement 1 to position the file pointer to the appropriate place, so that the data updation is done for the required item.
(ii) Write statement 2 to perform the write operation so that the updation is done in the binary file.

Ans. (i) file.seekp(-sizeof(Qty), ios::cur);
(ii) file.write((char*) &S, sizeof(stock));

Q. 10. Observe the program segment given below carefully and answer the questions that follow:

[Delhi, 2012]

class Inventory
{
    int Ano, Qty; char Article[20];
public:
    void Input () {cin>>Ano):
    gets(Article); cin>>Qty;}
    void Issue (int Q) {Qty+=Q;}
    void Procure (int Q) {Qty-=Q;}
}

int GetAno () {return Ano;}
void ProcureArticle (int TAno, int TQty)
{
    fstream File;
    File.open("STOCK.DAT", ios :: binary|ios :: in|ios :: out);
    Inventory I;
    int Found=0;
    while (Found==0 && File.read((char*) &I, sizeof(I)))
    {
        I. Procure (TQty);
        //Statement 1
        //Statement 2
        Found++;
    }
}

} if (Found==1)
    {cout<<" Procurement Updated" << endl;
    else
        cout<<"Wrong Article No"<<endl;
File.close ( );}

(i) Write statement 1 to position the file pointer to the appropriate place, so that the data updation is done for the required term.
(ii) Write statement 2 to perform the write operation so that the updation is done in the binary file.

Ans. (i) file.seekp(sizeof(qty), ios :: cur);
(ii) file.write((char*) &I, sizeof(I));

---

Short Answer Type Questions-I

Q. 1. Consider a file F containing objects E of class Emp.

[Delhi, 2015]

(i) Write statement to position the file pointer to the end of the file.

(ii) Write statement to return the number of bytes from the beginning of the file to the current position of the file pointer.

Ans. (i) F.seekg(0,ios::end);

(ii) F.tellg();

[½ mark for the statement]

[½ mark for the statement]

Q. 2. Write a function RevText() to read a text file “Input.txt” and Print only word starting with “T” in reverse order.

[Delhi, 2015]

Example : If value in text file is : INDIA IS MY COUNTRY

Output will be : AIDNI SI MY COUNTRY

Ans. void RevText()
{
    ifstream in ("Input.txt");
    char word[25];
    while(in)
    { in >>word;
        if (word[0] = = 'I')
            cout<<strrev(word);
        else
            cout<<word;
    }
}

[½ mark for opening the file in input mode]

[½ mark for reading a word]

[1 mark for displaying words starting with ‘T’ in reverse order]
Q. 3. Write a function in C++ to search display details, where destination is "Chandigarh" from binary file "Flight.Dat". Assuming the binary file is containing the objects of the following class:

```cpp
class FLIGHT
{
  int Fno; // Flight Number
  char From [20]; // Flight Starting Point
  char To[20]; //Flight Destination

  public:
    char * GetFrom(); { return from; }
    char * GetTo(); { return To; }

    void input() {cin>>Fno>>; gets(From); get (To);}
    void show() {cout<<Fno<<":""<<From""":""<<To<<endl;}
};

Ans. void Displaydetails()
{
  ifstream fin ("Flight.Dat");
  Flight F;
  while (fin)
  {
    fin.read((char*)&F,sizeof(F))
    if (strcmp(F.GetTo(), "Chandigarh")
      F.show();
  }
}
```

Any other correct function definition

½ Mark for opening WRITEUP.TXT correctly
½ Mark for reading each word (using any method) from the file
½ Mark for comparing the word with TOWER
½ Mark fo displaying correct count of TOWER
NOTE:
½ Mark to be deducted if TOWER is compared without ignoring the case

Q. 4. Write a definition for function TOWER() in C++ to read content of a text file WRITEUP.TXT. Count the presence of word TOWER and display the number of occurrences of this word.

```cpp
Ans. void TOWER()
{
  int count = 0;
  ifstream f ("WRITEUP.TXT") ;
  char s [20] ;
  while (! f .eof () )
  {
    f>>s;
    if (strcmp (s, "TOWER") == 0)
      count ++ ;
  }
  cout<<count;
  f.close ();
}
```

OR

Any other correct function definition

½ Mark for opening WRITEUP.TXT correctly
½ Mark for reading each word (using any method) from the file
½ Mark for comparing the word with TOWER
½ Mark for displaying correct count of TOWER
NOTE:
½ Mark to be deducted if TOWER is compared without ignoring the case

Q. 5. Write a definition for function COSTLY() in C++ to read each record of a binary file GIFTS.DAT, find and display those items, which are priced more than 2000. Assume that the file GIFTS.DAT is created with the help of objects of class GIFTS, which is defined below:

```cpp
class GIFTS
{
  int CODE; char ITEM [20]; float PRICE;
}
```

Note:
— The word TOWER should be an independent word
— Ignore type cases (i.e. lower/upper case)

Example:
If the content of the file WRITEUP.TXT is as follows:

```
Tower of hanoi is an interesting problem. Mobile phone tower is away from here. Views from EIFFEL TOWER are amazing.
```

The function TOWER() should display the following:

```
3
```
sizeof (G))
{
    if (G.GetPrice () >2000)
        G.View ();
}
fin. close ();

OR

Any other correct equivalent function definition
(½ Mark for opening GIFT.S.DAT correctly)
(1 Mark for reading all records from the file)
(1 Mark for checking value of PRICE > 2000)
(½ Mark for displaying the desired items)

Q. 6. Find the output of the following C++ code considering that the binary file MEMBER. DAT exists on the hard disk with records of 100 members:  [Delhi, 2015]
class MEMBER
{
    int Mno; char Name [20];
public :
    void In (); void out ();
};
void main ()
{
    ifstream MF;
    MF.open (:_MEMBER. DAT*, ios :: binary | ios :: in);
    MEMBER M;
    MF.read ({char*} &M, sizeof (M));
    MF.read ({char*} &N, sizeof (M));
    MF.read ({char*} &M, sizeof (M));
    int POSITION = MF. tellg () / sizeof (M);
    cout<<"PRESENT RECORD :"<<POSITION <<endl;
    MF.close ();
}

Ans. PRESENT RECORD : 3
(1 Mark for writing PRESENT RECORD : 3)
OR
(1 Mark for writing only 3)
OR
(½ Mark for writing only PRESENT RECORD :)

Q. 7. Write function definition for SUCCESS () in C++ to read the content of a text file STORY.TXT, count the presence of word STORY and display the number of occurrence of this word.  [O.D, 2015]

Note:
— The word STORY should be an independent word
— Ignore type cases (i.e. lower/upper case)
Example:
If the content of the file STORY.TXT is as follows

Success shows others that we can do it. It is possible to achieve success with hard work. Lot of money does not mean SUCCESS.

The function SUCCESS() should display the following:

3

Ans. void SUCCESS ()
{
    int count = 0;
    ifstream f ("STORY.TXT");
    char s [20];
    while (! f. eof () )
    {
        f>>s;
        if (strcmp (s, "STORY") == 0)
            // Or if (strcmp (s, "SUCCESS") == 0)
            count++;
    }
    cout<"count;
    f.close ();
}

OR

Any other correct function definition
(½ Mark for opening STORY.TXT correctly)
(½ Mark for reading each word (using any method) from the file)
(½ Mark for comparing the word with STORY OR SUCCESS)
(½ Mark for displaying correct count of STORY OR SUCCESS)
NOTE:
(½ Mark to be deducted if STORY or SUCCESS is compared without ignoring the case)

Q. 8. Write a definition for function Economic () in C++ to read each record of a binary file ITEMS.DAT, find and display those items, which costs less than 2500. Assume that the file ITEMS. DAT is created with the help of objects of class ITEMS, which is defined below:  [Delhi, 2015]
class ITEMS
{
    int ID; char GIFT [20]; float cost;
public :
    void Get ()
    {
        cin>>CODE; gets (GIFT);
        cin>>cost;
    }
    void See ()
    {
Q. 10. Write function definition for WORDABSTART () in C++ to read the content of a text file, JOY.TXT, and display all those words, which are starting with either 'A', 'a' or 'B', 'b'.

**Example:**
If the content of the file JOY.TXT is as follows:

I love to eat apples and bananas. I was travelling to Ahmedabad to buy some clothes.

The function WORDABSTART() should display the following:

apples bananas Ahmedabad buy

Ans. void WORDABSTART()
{
    ifstream fil;
    fil.open("JOY.TXT");
    char W[20];
    fil>>W;
    while(!fil.eof()) //OR While(fil)
    {
        if(strlen(W)==4)
            cout<<W<<" ";
        fil>>W;
    }
    fil.close(); //Ignore
}

(½ Mark for opening JOY.TXT)
(½ Mark for reading each word from the file)
(½ Mark for checking length of the word)
(½ Mark for showing the 4 letter extracted word correctly)

Q. 11. Write a definition for function OFFER() in C++ to read each object of a binary file OFFER.DAT, find and display details of those ITEMS, which has status as "ON OFFER". Assume that the file OFFER.DAT is created with the help of objects of class ITEMS, which is defined below:

```cpp
class ITEMS
{
    int ID; char Item[20]; Status[20];
    float Price;
}
```

Ans. void OFFER()
{
    ITEMS;
    ifstream fin;
    fin.open("OFFER.DAT", ios::binary);
    while(fin.read(char*) &G size of (G))
}
Short Answer Type Questions-II

Q.1. Given the binary file CAR.DAT, containing records of the following class CAR type:

```cpp
class CAR {
    int C_No;
    char C_Name[20];
    float Mileage;
    public:
    void enter() {
        cin>>C_No; gets(C_Name); cin >> Mileage;
    }
    void display() {
        cout<<C_No; cout<<C_Name ;
        cout<<Mileage;
    }
    int RETURN_Mileage() {
        return Mileage;
    }
};
```

Write a function in C++, that would read contents from the file CAR.DAT and display the details of car with mileage between 100 to 150.

Ans. void CARSearch()
{
    fstream FIL;
    FIL.open("CAR.DAT",ios::binary|ios::in);[1]
    CAR C;
    int Found = 0;
    while (FIL.read((char*)&C,sizeof(C))) [1]
    {
        if ((C.RETURN_Mileage() > 100) &&
            (C.RETURN_Mileage() < 150))
            { C.display();
              Found++;
            }
    }
    if (Found==0)
    cout<<"Sorry! No car found with mileage between 100 to 150";
    FIL.close();[1]
}

(1 Mark for opening OFFER.DAT correctly)
(½ Mark for reading record from OFFER.DAT)
(½ Mark for comparing Remarks with ON OFFER)
(½ Mark for showing record)

Q.2. Given a binary file “SPORTS.DAT” containing records of the following class:

```cpp
class Player {
    char PNO[10]; //player number
    char Name[20]; //Name of player
    int rank; //rank of the player
    public:
    void EnterData() {
        gets(PNO); gets(Name); cin>>rank;
    }
    void DisplayData() {
        cout<<setw(12)<<PNO;
        cout<<setw(32)<<Name;
        cout<<setw(3)<<rank<<endl;
    }
    int Ret_rank() {return rank;}
};
```

Write a function in C++ that would read contents of the file ‘SPORTS.DAT’ and display the details of those players whose rank is above 500.

Ans. void show()
{
    fstream file;
    file.open("SPORTS.DAT",ios::binary|ios::in);
    CAR C;
    int Found = 0;
    while (file.read((char*)&P,sizeof(P)))[1]
    {
        if(P.rank>500)
        { P.DisplayData();
          P.close();[1]
        }
    }

```
Q. 3. Assuming the class WORKER as declared below, write a function in C++ to read the objects of WORKER from binary file named WORKER.DAT and display those records of workers, whose Wage is less than 300.

```cpp
class WORKER
{
    int WNO;
    char WName[30]; float Wage;
public:
    void Enter() {cin>>WNO; gets (WName); cin>>wage;}
    void DISP() {cout<<ENo<<’ ’<<WName
                <<’ ’<<Wage<<endl;}
    float GetWage() {return Wage;}
};

Ans. void Display()
{
    ifstream fin;
    fin.open ("WORKER.DAT", ios : : binary);
    WORKER W;
    while (! fin. eof () )
    {
        fin. read ((char*) &W, sizeof (W));
        if (W. GetWage() <300)
            W. DISP ();
    }
    fin. close ();
}
```

[CBSE Compit., 2013]

Q. 4. Assuming the class TOYS as declared below, write a function in C++ to read the objects of TOYS from binary file TOYS.DAT and display those details of those TOYS, which are meant for children of AgeRange “5 to 8”.

```cpp
class TOYS
{
    int ToyCode;
    char ToyName[10]; char AgeRange;
public:
    void Enter()
    {
        cin>>ToyCode;
        gets (ToyName);
        gets(AgeRange);
    }
    void Display()
    {
        cout<<ToyCode
        ToyName<<endl;
        cout<<AgeRange<<endl;
    }
};
```

Q. 5. Assuming the class GAMES as declared below, write a function in C++ to read the objects of GAMES from binary file GAMES.DAT and display those details of those GAMES, which are meant for children of AgeRange “8 to 13”.

```cpp
class GAMES
{
    int GameCode;
    char GameName[10];
    char AgeRange;
public:
    void Enter()
    {
        cin>>GameCode;
        gets(GameName);
        gets(AgeRange);
    }
    void Display()
    {
        cout<<GameCode<<" : "<GameName<<endl;
        cout<<AgeRange<<endl;
    }
};
```

Q. 6. Assuming the class ANTIQUE as declared below, write a function in C++ to read the objects of Antique from binary file ANTIQUE.DAT and display those antique items, which are priced between 10000 and 15000.
class ANTIQUE
{
    int ANO;
    char Aname [10];
    float Price;
public :
    void BUY() {cin>>ANO;gets
    (Aname); cin>> price;} 
    void SHOW () {
        cout<<ANO<<endl;
        cout<<Aname<<endl;
        cout<<Price<<endl;
    }
    float Get Price () {return Price;}
}; [O.D. 2013]

Ans. void Disp() {
    ifstream fin;
    fin.open ("ANTIQUES.DAT", ios : : in | 
    ios : : out | ios : : binary);
    Antique A;
    while (!fin.eof()) {
        fin.read ((char*) &A, sizeof(A));
        if (A.GetPrice () >=10000 & 
            A.GetPrice() <15000)
            A.SHOW( );
    }
    fin.close ( );
} [3]

Q. 7. Assuming the class VINTAGE as declared below,
write a function in C++ to read the objects of
VINTAGE from binary file VINTAGE.DAT and
display those vintage vehicles, which are priced
between 200000 and 250000.

class VINTAGE
{
    int VNO; //Vehicle Number
    char VDesc[10]; //Vehicle
    Description
    float Price;
public:
    void GET( ) {cin>>VNO;gets
    (VDesc); cin>>Price,}
    void VIEW( ) {
        cout<<VNO<<endl;
        cout<<VDesc<<endl;
        cout<<Price<<endl;
    }
    float ReturnPrice ( ) {return
    Price;}
};

Ans. void Disp( ) {
    ifstream fin;
    fin.open ("VINTAGE.DAT", ios : : in |
    ios : : binary);
    vintage V;
    while (!fin.eof()) {
        fin.read ((char*) &V, sizeof(V));
        if (V.ReturnPrice () > 200000 & 
            V.ReturnPrice() <250000)
            V.VIEW ( );
    }
    fin.close ( );
} [3]

Q. 8. Write a function display () in C++ to display all
the students who have got a distinction (scored
percentage more than or equal to 75) from a
binary file *stud.dat", assuming the binary file is
containing the objects of the following class:

class Student
{
    int rno;
    char name [20];
    int percent;
    public;
    int retpercent()
    {
        return percent;
    }
    void getdetails() {
        cin>>rno;
        gets(sname);
        cin>>percent;
    }
    void showdetails() {
        cout<<rno;
        puts(sname);
        cout<<percent;
    }
};

Ans. void display() {
    student s;
    ifstream i("stud.dat");
    while(i.read(char*) &s, sizeof(s)))
    {
        if(s.retpercent()>=75)
            s.showdetails();
    }
    i.close();
}

(½ Mark for opening stud.dat correctly)
(1 Mark for reading all records from the file)
(1 Mark for comparing desired value with obtained
data)
(½ Mark for calling showdetails() function)

[CBSE Marking Scheme, 2016]
Q. 9. Write a function in C++ to search for the details (Phone no. and Calls) of those phones, which have more than 800 calls from a binary file "phones.dat". Assuming that this binary file contains records/objects of class phone, which is defined below.

```cpp
#include <iostream>
#include <fstream>

class Phone {
    char Phonenumber[10];
    int Calls;
    
    public:
        void Get() {
            cout << "Enter phone number: " << Phonenumber << endl;
            cin >> Phonenumber;
        }
        void Billing() {
            cout << "Enter number of calls: " << endl;
            cin >> Calls;
            if (Calls > 800) {
                cout << "Billing successful." << endl;
                // Add billing logic here
            } else {
                cout << "Billing failed." << endl;
            }
        }
};
```

Ans. void Search()
```cpp
ifstream f;
    f.open("mobile.dat");
    while (!f.eof()) {
        for (int i = 0; i < 10; i++) {
            if (f.read((char*) &m[i], sizeof(m[i]))) {
                if (m[i].GetCalls() > 800) {
                    Phone phone;
                    phone.Get();
                    phone.Billing();
                }
            }
        }
    }
``` [0D, 2012]

Q. 11. Write a function in C++ to search for a camera from a binary file "CAMERA.DAT" containing the objects of class Camera (as defined below). The user should enter the model no and the function should search and display the details of the camera.
```cpp
class Camera {
    long ModelNo;
    float Mm; float Zoom;
    char Details[120];

    public:
        void Read() {
            cin >> ModelNo >> Mm >> Zoom;
            gets (Details);
        }
        void Display() {
            cout << "Model No: " << ModelNo << " Mm: " << Mm << " Zoom: " << Zoom << " Details: " << Details << endl;
        }
        long GetModelNo() {return ModelNo;}
};
```

Ans. void cameraSearch()
```cpp
ifstream FIL; FIL.open("CAMERA.DAT", ios::binary|ios::in);
Camera B;
    long bn; int Found = 0;
    cout << "Enter the Model Number";
    cin >> bn;
    while ((FIL.read((char*) &B, sizeof(B))) != EOF) {
        if (FIL.GetModelNo() == bn) {
            B.Display();
            Found ++;
        }
    }
    if (Found == 0) {
        cout << "Sorry! Camera not found" << endl;
        FIL.close();
    }
``` [3]

**Long Answer Type Questions** [4 marks each]

1. Write function definition for WORD4CHAR() in C++ to read the content of a text file FUN.TXT and display all those words, which have four characters in it.

Ans. #include <iostream>
```cpp
int main() {
    std::ifstream file("FUN.TXT");
    std::string word;
    while (std::getline(file, word)) {
        if (word.length() == 4) {
            std::cout << word << std::endl;
        }
    }
    return 0;
}
``` [O.D, 2016]
When I was a small child, I used to play in the garden with my grand mom. Those days were amazingly fun and I remember all the moments of that time.

The function `WORD4CHAR()` should display the following:

```
When used play with days were that time.
```

**Ans.**

```c
void WORD4CHAR()
{
    ifstream f("FUN.TXT");
    char ch[20];
    while (!f.eof())
    {
        f >> ch;
        if(strcmp(ch, "days") == 0)
        {
            cout << ch;
        }
    }
    f.close();
}
```

**Q.2.** Write a definition for function `ONOFFER(fin)` in C++ to read each object of a binary file TOYS.DAT, find and display details of those toys, which have status as "ON OFFER". Assume that the file TOYS.DAT is created with the help of objects of class TOYS, which is defined below:

```
Class TOYS
{
    int TID; char Toy[20], Status[20]; float MRP;
    public:
        void Getinstock()
        {
            cin >> TID; gets(Toy); gets(Status); cin >> MRP; }
        void View()
        {
            cout << "TID:" << TID << "Toy:" << Toy << "Status:" << Status << endl;
        }
    char *SeeOffer() {Return Status;}
}
```

Ans. **ONOFFER()**

```
    void ONOFFER()
    {
        ifstream fin;
        fin.open("TOYS.DAT", ios :: binary);
        while (fin.read((char *)&T, size(T)))
        {
            if (strcmp(T.see offer(), "ON OFFER") == 0)
                T.view();
        }
        fin.close(); // Ignore
    }
```

**Q.3.** Find the output of the following C++ code considering that the binary file CLIENT.DAT exists on the hard disk with a data of 1000 clients:

```
class CLIENT
{
    int Ccode; char CName[20];
    public:
        void Register(); // void Display();
};

void main()
{
    ifstream CFile;
    CFile.open("CLIENT.DAT");
    CLIENT C;
    CFile.read((char *)&C, sizeof(C));
    cout << "Rec:" << CFile.tellg() /
         sizeof(C) << endl;
    CFile.read((char *)&C, sizeof(C));
    CFile.read((char *)&C, sizeof(C));
    CFile.read((char *)&C, sizeof(C));
    cout << "Rec:" << CFile.tellg() /
         sizeof(C) << endl;
    CFile.close();
}
```

Ans. **Rec:**

```
Rec: 1
Rec: 3
```

**Know the Links**

Download Chapterwise notes, study materials, Ten Year Solved Papers, Important Questions of Physics, Chemistry, Maths, Biology, Computer Science from www.edusuccess.in