**Phonix Group (Group 5)**

**Data File Handling In C++**

**Question Bank**

**For Slow Bloomers**

1. **How are binary files different from text files in C++?**

**Answer** : A text file store information in ASCII characters. In text files, each line of text is terminated, with a special character known as EOL character.   
A binary file store information in the same format in which the information is held in memory. In binary file, there is no delimeter for a line.

**2. What is a stream? Name the streams generally used for file I/O.**

Answer : A stream is a sequence of byte.   
ofstream: Stream class to write on files   
ifstream: Stream class to read from files   
fstream: Stream class to both read and write from/to files.

**3. Difference between get() and getline().**

Answer : get() does not extract the delimeter newline character from input stream. On the other hand getline() does extract the delimeter newline character from the input stream so that the stream is empty after getline() is over.

**4. Difference between ios::app and ios::out.**

Answer : The ios::out is the default mode of ofstream. With the mode of the file does not exist, it gets created but if the file exists then its existing contents get deleted.  
The ios::app is output mode of ofstream. With the mode of the file does not exist, it gets created but if the file exists then its existing contents are retained and new information is appended to it.

**5. Discuss the files stream classes defined inside fstream.h header file.**

**Answer .**ifstream: can be used for read operations.

ofstream: can be used for write operations.

fstream: can be used for both read & write operations.

**6. 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.**

**#include<fstream.h>**

**class PRODUCT{**

**int Pno;**

**char Pname[20];**

**int Qty;**

**public:**

**void ModifyQty(); //the function is to modify quantity of a PRODUCT**

**};**

**void PRODUCT::ModifyQty()**

**{**

**fstream File;**

**File.open(“PRODUCT.DAT”,ios::binary|ios::in|ios::out);**

**int MPno;**

**cout<<”product no to modify quantity:”;**

**cin>>MPno;**

**while(File.read((char\*)this,sizeof(PRODUCT)))**

**{**

**if(MPno==Pno)**

**{**

**cout<<”present quantity:”<<Qty<<endl;**

**cout<<”changed quantity:”;**

**cin>>Qty;**

**int Position= ; //statement 1**

**; //statement 2**

**File.write((char\*)this,sizeof(PRODUCT)); //Re-writing the record**

**}**

**}**

**File.close();**

**}**

**Ans.**Statement 1:

int Position=File.tellg( );

Statement 2:

File.seekp(Position-sizeof(PRODUCT),ios::beg);

**7. Observe the program segment given below carefully and fill the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task.**

**#include<fstream.h>**

**class Stock{**

**long Ino; // Item Number**

**char Item[20]; // Item Name**

**int Qty; // Quantity**

**public:**

**void Get(int);**

**Get(int);// Function to enter the content**

**void Show( ); // Function to display the content**

**void Purchase(int Tqty)**

**{**

**Qty+ = Tqty; // Function to increment in Qty**

**}**

**long KnowIno( )**

**{ return Ino;}**

**};**

**void Purchaseitem(long PINo, int PQty)**

**// PINo -> Info of the item purchased**

**// PQty -> Number of items purchased**

**{**

**fstream File;**

**File.open(“ITEMS.DAT”,ios::binary|ios::in|ios::cut); int Pos=-1;**

**Stock S;**

**while (Pos== -1 && File.read((char\*)&S, sizeof(S)))**

**if (S.KnowInc( ) == PINo)**

**{**

**S.Purchase(PQty); // To update the number of items**

**Pos = File.tellg()- sizeof(S);**

**//Line 1 : To place the file pointer to the required position**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;**

**//Line 2 : To write the objects on the binary file**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;**

**}**

**if (Pos == -1)**

**cout<<“No updation done as required Ino not found…”;**

**File.close( );**

**}**

**Ans.**Line 1:

File.seekp(Pos);

Line 2:

File.write((char\*) &S, sizeof(S));

**8. 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;

}

**9. 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);

}

**10. Given a binary file TELEPHONE.DAT, containing records of the following class Directory**

class Directory

{ char name[20],address[30], areacode[5], 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 all those records having areacode as “123” from TELEPHONE.DAT to TELEBACK.DAT**

COPYABC()

{ifstream ifile(“TELEPHONE.DAT”,ios::in|ios::binary);

If(!ifle) { cout<<”could not open TELEPHONE.DAT”; exit(-1);}

else

{ofstream ofile(“TELEBACK”,ios::out|ios::bainary);

if(!ofile) {cout<<”could not open TELEBACK.DAT”; exit(-1);}

else

{Directory d;

while(ifile.read((char \*)&d, sizeof(d)))

{if(d.checkcode(“123”)==0)

Ofile.write((char \*)&d,sizeof(d));

}

ifile.close();

ofile.close();

}

}

}

**For Bloomers**

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

**class PracFile**

**{ int Pracno;**

**char PracName[20];**

**char TimeTaken;**

**int Marks;**

**public:**

**//function to enter PracFile details**

**void EnterPrac();**

**//function to display PracFile details**

**void ShowPrac();**

**//function to return TimeTaken**

**char RTime() { return TimeTaken; }**

**//fuction to assign Marks**

**void Assignmarks(int M)**

**{ Marks=M;**

**};**

**};**

**void AllocateMarks()**

**{ fstream File;**

**File.open(“MARKS.DAT”,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 file MARKS.DAT based on their value of 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 statement 2 is to perform the write**

**operation with the modified record.**

**Ans.**Statement 1: File.seekp((Record)\*sizeof(P));

OR

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

Statement 2: File.write((char\*)&P,sizeof(P));

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

**class Team**

**{ long TId[10]; //Team’s Id**

**char TName[20]; //Team’s Name**

**float Points; //Team’s Points**

**public:**

**void Accept();**

**void Show();**

**void PointChange(); //Function to change Points**

**long R\_TId() {return TId; }**

**};**

**void ReplacePoints(long Id)**

**{ fstream File;**

**File.open(“TEAM.DAT”,ios::binary|ios::in|ios::out);**

**Team T;**

**int Record=0;Found=0;**

**while(!Found && File.read((char\*)&T,sizeof(T)))**

**{**

**if(Id==T.R\_TId())**

**{ cout<<“Enter new Points”;**

**T.PointsChange();**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //Statement 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //Statement 1**

**Found=1;**

**}**

**Record++;**

**}**

**if(found==1)**

**cout<<“Record Updated”;**

**File.close();**

**}**

**Write the statement 1 to position the File Pointer at the beginning of the Record for which the Team’s Id matches with the argument passed, ad statement 2 to write the updated Recode at that position.**

**Ans.**Statement 1:

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

Statement 2:

File.write((char\*)&T,sizeof(T));

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

**class Candidate**

**{ long CId; //Candidate’s Id**

**char CName[20]; //Candidate’s Name**

**float Marks; //Candidate’s Marks**

**public:**

**void Enter();**

**void Display();**

**void MarksChange();**

**//Funcion to change marks**

**long R\_CId() { return CId; }**

**};**

**void MarksUpdate(log ID)**

**{ fstream File;**

**File.open(“CANDIDATE.DAT”,ios::binary|ios::in|ios::out);**

**Candidate C;**

**int Record=0,Found=0;**

**while(!Found&&File.read((char\*)&C,sizeof(C)))**

**{ if(Id==C.R\_CId())**

**{ cout<<“Enter new Marks”;**

**C.MarkChange();**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //Statement 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //statement 2**

**Found=1;**

**}**

**Record++;**

**}**

**if(found==1) cout<<“Recoed Updated”;**

**File.close();**

**}**

**Write the statement 1 to position the File Pointer at the beginning of the Record for which the Candidate’s Id matches with the argument passed, ad statement 2 to write the updated Recode at that position.**

**Ans.**Statement 1:

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

Statement 2:

File.write((char\*)&C,sizeof(C));

**4. What are input and output streams? What is the significance of fstream.h file?**

**Ans.** Input stream: The stream that supplies data to the program is known as input stream.

Output stream: The stream that receives data from the program is known as output stream. fstream.h file includes the definitions for the stream classes ifstream, ofstream and fstream. In C++ file input output facilities implemented through fstream.h header file.

**5. What role is played by file modes in file operations? Describe the various file mode constants and their meanings.**

**Ans.** A file mode describes how a file is to be used: read it, write to it, append it, and so on. Different file modes constants and their meanings are as following:

**Constant Meaning**

ios::in               Opens file for reading.

ios::out            Opens file for writing.

ios::ate             This seeks to end-of-file upon opening of the file.

ios::app            This causes all output to that file to be appended to the end.

ios::trunc         The contents of a pre-existing file by the same name to be destroyed and truncates the file to zero length.

ios::nocreate   Causes open() function to fail if the file does not already exist.

ios::noreplace  Causes open() function to fail if the file already exist.

ios::binary       Causes a file to be opened in binary mode.

6. **What are the steps involved in using a file in a C++ program?**

**Ans.** In order to process files, follow these steps:

(i) Determine the type of link.

(ii) Declare a stream accordingly.

(iii) Link file with the stream

(iv) Process as required, and

(v) De-link the file with the stream.

7. **Describe the various classes available for file operations.**

**Ans. Class Functions**

filebuf It sets the file buffers to read and write.

fstreambase This is the base class for fstream, ifstream and ofstream classes.

ifstream It provides input operations for file.

ofstream It provides output operations.

fstream It provides support for simultaneous input and output operations.

8. **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;

}

**9. 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;

}

10. Given a binary file GAME.DAT, containing records of the following structure type

struct Game  
{ char GameName[20] ;  
char Participate[10][30] ;  
} ;

Write a function in C++ that would read contents from the file GAME.DAT and creates a file named BASKET.DAT copying only those records from GAME.DAT where the game name is “Basket Ball”.

**Solution:**

void BPlayers( )  
{ ifstream  
fin(“GAME.DAT’,ios::in,ios::binary););  
ofstream fout(“BASKET.DAT”,ios::out|  
ios::binary);  
Game G;  
while(fin) // or while(!fin.eof( ))  
{ fin.read((char\*)&G,sizeof(Game));  
if(strcmp(G.GameName,”Basket Ball”)= = 0)  
fout.write((char\*)&G,sizeof(G));  
}  
fin.close( );  
fout.close( );  
}

**High Achivers**

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

**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**

**it 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\_OB())**

**{ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //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 fileBOOK.DAT with the values of Book NEW passed to**

**its argument, write the appropriate statement for Missing Statement using seekp() or seekg(), whichever needed,**

**in the above code that would write the modified record at its proper place.**

**Ans.**File.seekg(-1\*sizeof(NEW),ios::cur);

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

**#include<fstream.h>**

**class Customer**

**{**

**long Cno;**

**char Name[20],Mobile[12];**

**public:**

**//function to allow user to enter the Cno, Name, Mobile**

**void Enter();**

**//function to allow user to enter (modify) mobile number**

**void Modify();**

**//function to return value of Cno**

**long GetCno() { return Cno;}**

**};**

**void ChangeMobile()**

**{**

**Customer C;**

**fstream F;**

**F.open(“CONTACT.DAT”,ios::binary|ios::in|ios::out);**

**long Cnoc; //customer no. whose mobile number needs to be changed**

**cin>>Cnoc;**

**while(F.read((char\*)&C,sizeof(c)))**

**{**

**If(Cnoc==C.GetCno())**

**{**

**C.Modify();**

**//statement 1**

**Int Pos= \_\_\_\_\_\_\_\_\_\_\_\_\_\_ //to find the current position**

**//of file pointer**

**// statement 2**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //to move the file pointer to write the**

**//modified the record back on to the file**

**//for the desired Cnoc**

**F.write((char\*)&C,sizeof(c));**

**}**

**}**

**File.close();**

**}**

**Ans.**Statement 1:

F.tellg() ;

Statement 2:

F.seekp(Pos-sizeof(C));

OR

F.seekp(-l\*sizeof(C) ,ios::cur);

3. **Discuss the two methods of opening a file within a C++ program. When is one method preferred over the other?**

**Ans.** A file can be opened in two ways :-

**a) Using the constructor of the stream class –**This method is useful when only one file is used in the stream.

Constructors of the stream classes ifstream, ofstream and fstream are used to initialize the file stream object with the file name. For example,

ifstream read\_file(“Names.Dat”);

**b) Using the function open() –**This method is useful when we want to use different files in the stream. If two or more files are to be processed simultaneously, separate streams must be declared for each. For example, ifstream ifl; //input stream ifl created ifl.open(“Names.Dat”); // file Names.Dat linked with ifl Second method is preferred over first method when there is a situation to open more than one file.

**4.. Both ios::ate and ios::app place the file pointer at the end of the file when it is opened. What then, is the difference between them?**

**Ans.**Both ios::ate and ios::app place the file pointer at the end of the file when it is opened. The difference between the two is that ios::app lets you add data to the end of the file only, while the ios::ate mode when opened with ofstream allows you to write data anywhere in the file, even over old data.

**5. When a file is opened for output what happens when**

**(i) the mentioned file does not exist.**

**(ii) the mentioned file does exist.**

**Ans.**(i) Creates a new file.

(ii) the act of opening a file for output scraps it off so that output starts with a fresh file.

**6. Suggest the situation where write() and read() are preferred over get() and put() for file I/O operations. Support your answer with examples.**

**Ans.** The get() and put() functions perform I/O byte by byte. On the other hand, read() and write() functions let you read and write structures and objects in one go without creating need for I/O for individual constituent fields.

Example:

file.get(ch);

file.put(ch);

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

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

**7. Discuss the working of good() and bad() functions in file I/O error handling.**

**Ans.** good(): Returns nonzero (true) if no error has occurred. For instance, if fin.good() is true, everything is okay with the stream named as fi and we can proceed to perform I/O operations. When it returns zero, o further operations can be carried out.

bad(): Returns true if a reading or writing operation fails. For example in the case that we try to write to a file that is not open for writing or if the device where we try to write has no space left.

**8. A program to display the size of a file in bytes.**

#include<iostream.h>

#include<fstream.h>

#include<process.h>

#include<conio.h>

int main()

{

char filename[13];

clrscr();

cout<”Enter Filename:\n”;

cin.getline(filename,13);

ifstream infile(filename);

if(!infile)

{cout>>”sorry ! Can not open “<<filename <<”file\n”;

exit(-1);

}

long no\_bytes=0;

char ch;

infile.seekg(0,ios::end);

no\_bytes=infile.tellg();

cout<<”File Size is”<<no\_bytes<<”bytes\n”;

return 0;

}

9. C++ program, which initializes a string variable to the content “There is an island of opportunity in the middle of every difficulty.” and output the string one character at a time to the disk file “OUT.TXT”.

#include<fstream.h>

int main()

{ ofstream fout(“OUT.TXT”);

char \*str = ”There is an island of opportunity in the middle of every difficulty.” ;

int i=0;

if(!fout)

{

cout<<”File cannot be opened “;

return 0;

}

while (str[i]!=’\0’)

{fout<<str[i];

i++;

}

fout.close();

}

**10. Function to add more objects belonging to class JOKE at the end of JOKES.DAT file.**

class JOKE{int jokeid; char type[5], jokedesc[200];

public:

void Newjokeentry(){cin>>jokeid>>type; cin.getline(jokedesc,200);}

void showjoke(){cout<<jokeid<<”\t”<<type<<endl<<jokedesc<<endl;}

};

void append()

{

fstream afile;

afile.open(“JOKES.DAT”, ios::binary | ios::app);

JOKE j;

int n,i;

cout<<”How many objects you want to add :”;

cin>>n;

for (i=0;i<n;i++)

{

j.Newjokeentry();

afile.write((char \*)& j, sizeof (JOKE));

}

afile.close();

}