Compiled By:
Er. RANJAN RAJ ARYAL
Amrit Campus

Unit 8 File Handling [6Hrs]

INTRODUCTION

All programs require some input and produce some output but those input and output are lost as the program terminates. Files are required to save our data for future use. Programs would not be very useful if they cannot store data in files. When data volume is large it is generally not convenient to enter the data through console. In case cases data can be stored in a file and then the program can read the data from data file rather than from the console.

The various operations possible on a data file using C++ programs are:

1. Opening a file
2. Reading data stored in the file
3. Writing/appending data from a program into a data file
4. Saving the data file onto some secondary storage device
5. Closing the data file once the ensuring operations are over
6. Checking status of file operation

Stream Class Hierarchy

Opening a file

The first operation generally done on an object of one of the stream classes is to associate it to a real file, that is to say, to open a file. The open file is represented within the program by a stream object and any input or output performed on this stream object will be applied to the physical file. A data file can be opened in a program in many ways. These methods are described below:

Ifstream filename("filename<with path>");

OR

ofstream filename("filename<with path>");

Closing a file

When reading, writing or consulting operations on a file are complete we must close it to so that it becomes available again. In order to do that we shall call the member function close (), that is in charge of flushing the buffers and closing the file. Its form is quite simple:

void close()

Once this member function is called, the stream object can be used to open another file and the file is available again to be opened by other processes.

Unformatted Input/output

Reading Data by getLine () Method:

With extraction operator reading terminates after reading white space character therefore above program is able to read single word from file. We can overcome above problem by using getline () function as below:

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
	ifstream fin;
	char str[80];
	fin.open("G:\\test.txt");
	fin.getline(str,79);
	cout<<"Reading contents From file:\n"<<str<<endl;

	return 0;
}
```

```
Reading contents From file:
----!!!This is a test file for File Handling.-------!!!!!
```

Using get and put Methods and Detecting End of File

The class ifstream has a member function eof() that returns a nonzero value if the end of file has been reached. This value indicates that there are no more characters in the file to be read further. This function is therefore used in the while loop for stopping condition. End of file can be detected in two ways: Using EOF () member function and using filestream object.

Detecting End of File using EOF() member function

```
#include<iostream>
#include<fstream>
using namespace std;
```

```
int main()
{
	ifstream fin;
	char ch;
	fin.open("G:\\test.txt");
	while(!fin.eof())
	{
		fin.get(ch);
		cout<<ch;
	}

	return 0;
}
```

```
----!!!This is a test file for File Handling.-------!!!!!


HELLO AGAIN
--------------------------------
```

Detecting End of File using Filestream Object

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
	ifstream fin;
	char ch;
	fin.open("G:\\test.txt");
	while(fin)
	{
		fin.get(ch);
		cout<<ch;
	}

	return 0;
}
```

```
----!!!This is a test file for File Handling.-------!!!!!


HELLO AGAIN
--------------------------------
```

Write a program to read the contents of a file and display them on the screen insertion opereator and getline method:

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
	ifstream fin;
	char str[100];
	fin.open("G:\\test.txt");
	while(!fin.eof())
	{
		fin.getline(str,79);
		cout<<str;
	}
	return 0;
}
```

```
----!!!This is a test file for File Handling.-------!!!!!HELLO AGAIN
--------------------------------
```

Write a program to read the contents of a text file and display them on the screen using extraction operator

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
	ifstream fin;
	char str[100];
	fin.open("G:\\test.txt");
	while(!fin.eof())
	{
		fin>>str;
		cout<<str<<" ";
	}

	return 0;
}
```

```
----!!!This is a test file for File Handling.-------!!!!! HELLO
--------------------------------
```

Reading and Writing by Using Read() and Write Member Functions

Files streams include two members functions specifically designed to input and output binary data sequentially: write and read. The first one (write) is a member function of ostream class inherited by ofstream and read is a member function of isstream class that is inherited by ifstream. Objects of class fstream have both members. Their prototypes are:

Syntax for write()

FileObject.write ((char*)&object,sizeof(object))

Syntax for read()

FileObject.read ((char*)&object,sizeof(object))

Example:

```
/*-------- Program to write data by using write() member function --------*/
#include<iostream>
#include<fstream>
using namespace std;
class Student
{
        public:
        int roll;
        char name[20];
        char address[20];
};
int main()
{
        Student s;
        ofstream fout;
        fout.open("G:\\student.txt");
        cout<<"Enter Rollno:"<<endl;
        cin>>s.roll;
        cout<<"Enter Name:"<<endl;
        cin>>s.name;
        cout<<"Enter Address:"<<endl;
        cin>>s.address;
        fout.write((char *)&s,sizeof(Student));
        fout.close();
        cout<<"Writing complete"<<endl;
        return 0;
```

}

```
Enter Rollno:
112
Enter Name:
Ramu
Enter Address:
Kathmandu
Writing complete
```

Program to read data from a binary File using read() member function

```
#include<iostream>
#include<fstream>
using namespace std;
class Student
{
        public:
        int roll;
        char name[20];
        char address[20];
};
int main()
{
        Student s;
        ifstream fin;
        fin.open("G:\\student.txt");
        fin.read((char *)&s,sizeof(Student));
        cout<<"Rollno:"<<s.roll<<endl;
        cout<<"Name:"<<s.name<<endl;
        cout<<"Address:"<<s.address<<endl;
        fin.close();
        cout<<"Reading complete"<<endl;

        return 0;
}
```

```
Rollno:112
Name:Ramu
Address:Kathmandu
Reading complete
```

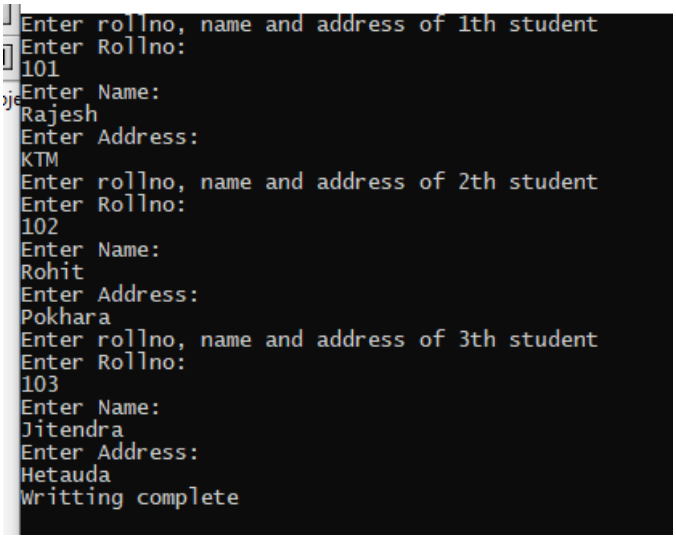
Writing multiple objects to file

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

```
using namespace std;
class Student
{
        private:
                        int roll;
                        char name[20];
                        char address[20];
    public:
                void read_data()
                {
                                cout<<"Enter Rollno:"<<endl;
                 cin>>roll;
                 cout<<"Enter Name:"<<endl;
                 cin>>name;
                 cout<<"Enter Address:"<<endl;
                 cin>>address;
                }
                void write_data()
                {

                }

};
int main()
{
        Student s;
        ofstream fout;
        fout.open("G:\\student.txt");
        for(int i=1;i<=3;i++)
        {
           cout<<"Enter rollno, name and address of "<<i<<"th student"<<endl;
                s.read_data();
           fout.write((char *)&s,sizeof(Student));
   }
        fout.close();
        cout<<"Writting complete"<<endl;
        return 0;
}
```



Reading multiple objects from file

```
#include<iostream>
#include<fstream>
using namespace std;
class Student
{
        private:
                int roll;
                char name[20];
                char address[20];
        public:
                void write_data()
                {
                        cout<<roll<<"\t"<<name<<"\t"<<address<<endl;
                }

};
int main()
{
        Student s;
        ifstream fin;
        fin.open ("G:\\student.txt");
        cout<<"Rollno\tName\tAddress"<<endl;
```

```
        for(int i=1;i<=3;i++)
        {
           fin.read ((char *)&s,sizeof(Student));
           s.write_data();
   }
        fin.close();
        cout<<"Reading complete"<<endl;
        return 0;
}
```

```
Rollno  Name    Address
101     Rajesh  KTM
102     Rohit   Pokhara
103     Jitendra        Hetauda
Reading complete
```

Random Access File Access

In some situations, you might want to read some record randomly not sequentially. You can do this using two models: One uses an absolute location in the stream called the streampos; the second woks like the standard C library functions **fseek ()** for a file and moves a given number of bytes from the beginning, current and end part of file.

The streampos approach requires that you first call a "tell" function; tell() for an ostream or tellg() for an istream. (The "p" refers to the "put pointer" and "g" refers to the "get pointer"). This function returns a streampos which you can later use in calls to seekp () for an ostream or seekg () for an stream. The second approach is a relative seek and uses overloaded versions of seekp() and seekg(). The first argument is the number of characters to move, it can be positive or negative.
The second argument is the seek direction. Some important member functions are:
seekg(): It is used to move reading pointer forward and backward.
Syntax:
Fileobject.seekg (50,ios::cur);// Moves 50 bytes forward from current
Fileobject.seekg (50,ios::beg);// Moves 50 bytes forward from beginning

Fileobject.seekg (50,ios::end);// Moves 50 bytes forward from end

seekp ():-It is used to move writing pointer forward and backward
Syntax:
Fileobject.seekp (no_of_bytes, mode)
fout.seekp (50,ios::cur)// Moves 50 byes forward from current position
fout.seepk (50, ios::beg) // Moves 50 bytes forward from current position

fout.seekp (50,ios::end);// Moves 50 bytes forward from end

tellp():- It returns the distance of writing pointer from the beginning in byes
Syntax:
FileObject.tellp ()
Example:
long n = fin.tellp ();
tellg():- It returns the distance of reading pointer from the beginning in bytes.
Syntax:
FileObject.tellg ()
Example:
Long n = fout.tellg ();

Example:

```
/*--------Program to read third object from the file student.txt--------*/
#include<iostream>
#include<fstream>
using namespace std;
class student
{
        int roll;
        char name[20];
        char address[20];
        public:
        void display()
        {
                cout<<"Rollno:"<<roll<<endl;
                cout<<"Name:"<<name<<endl;
                cout<<"Address:"<<address<<endl;
        }

};
int main()
{
        student s;
        int i;
        ifstream fin;
        fin.open("G:\\student.txt");
```

```
        fin.seekg(sizeof(s)*2,ios::cur);
        fin.read((char*)&s,sizeof(student));
        s.display();
        fin.close();
        return 0;
}
```

```
Rollno:103
Name:Jitendra
Address:Hetauda
```

/*----------Update/Modify the content of the file---------- */

```
#include<iostream>
#include<fstream>
#include<string.h>
#define N 2
using namespace std;
class Student
{
  public:
       int rollno;
       char name[20];
       char address[20];
      void read_data()
      {
          cout<<"Enter rollno"<<endl;
          cin>>rollno;
          cout<<"Enter name"<<endl;
          cin>>name;
          cout<<"Enter address"<<endl;
          cin>>address;
      }
      void display()
      {

         cout<<rollno<<"\t"<<name<<"\t"<<address<<endl;

      }
};
```

```
int main()
{
   Student s;
   fstream fin;
   int i,r;
   fin.open("G:\\student.txt");
   cout<<"Reading student information"<<endl;
   cout<<"Rollno\tName\tAddress"<<endl;
   while(fin.read((char *)&s, sizeof(Student)))
   {
       s.display();
   }
   if(fin.eof())
   fin.clear();
   cout<<"\nEnter the rollno of student whose record is to be modified"<<endl;
   cin>>r;
   fin.seekp(sizeof(s)*(r-1));
   cout<<"Enter new record"<<endl;
   s.read_data();
   fin.write((char *)&s,sizeof(Student));
   fin.seekg(0);
   cout<<"The modified record"<<endl;
   while(fin.read((char *)&s, sizeof(Student)))
   {
     if(strcmp(s.address,"KTM")==0)
      s.display();
   }
return 0;
}
```

```
Reading student information
Rollno  Name    Address
1       Rajesh  KTM
2       Rohit   KTM
3       Rekha   PKH

Enter the rollno of student whose record is to be modified
1
Enter new record
Enter rollno
RRR
Enter name
Enter address
The modified record
2       Rohit   KTM
```

/*---------- Program to Count Number of objects from file----------*/

```
#include<conio.h>
#include<iostream>
#include<fstream>
#define N 2
using namespace std;
class Student
{
  private:
        int rollno;
        char name[20];
        char address[20];
  public:
       void read_data()
       {
          cout<<"Enter rollno"<<endl;
          cin>>rollno;
          cout<<"Enter name"<<endl;
          cin>>name;
          cout<<"Enter address"<<endl;
          cin>>address;
       }
};
int main()
{
   Student s;
   ofstream fout;
```

```
    int i;
    fout.open("d:\\abc\\student.dat",ios::app);
    cout<<"Enter details of "<<N<<"students"<<endl;
    for(i=1;i<=N;i++)
    {
      s.read_data();
      fout.write((char *)&s,sizeof(Student));
    }
    int end = fout.tellp();
    int ob = end/sizeof(s);
    cout<<"Number of objects = "<<ob;
    fout.close();
    getch();
    return 0;
}
```