

File handling

Lecture 22

Files

- Storage of data in variables is temporary
- Permanent retention of data – file
- Secondary storage – CD, hard disk, pen drive etc.

Data

- Field – a group of characters that conveys meaning
- Record – composed of many fields.
 - To facilitate the retrieval of specific records from a file, at least one field is chosen as *record key*

Record organization

- Several ways of organizing records in a file
 - Sequential files where records are stored in order by record-key field

Databases

- Applications use many different files to store data
- A group of such related files is called database

Streams

- C++ input/output occurs in streams
- Sequence of bytes
- Input operation – bytes flow from input devices into main memory
- Output operation – bytes flow from main memory into output devices

Streams (contd..)

- Application associates meaning with bytes
- System I/O transfers bytes from devices to memory and vice versa
- C++ provides
 - Low level input/output (unformatted)
 - High level input/output (formatted)

Files and streams

- C++ views files as sequence of bytes
- Each file ends either with end-of-file marker or at a specific byte number system-maintained, administrative data structure

Files and streams (contd..)

- When a file is opened, an object is created and a stream is associated with the object
- For example, objects cin, cout are created when <iostream> is included
 - Streams associated with these objects provide communication channels between program and device

File handling

- To perform file processing in C++, header file `<iostream>` and `<fstream>` must be included
- `<fstream>` includes definitions for the stream class templates
 - `basic_ifstream` (for file input)
 - `basic_ostream` (for file output)
 - `basic_fstream` (for file input/output)

Example

```
#include<iostream>
#include<fstream>
using namespace std;
void main( )
{ ofstream out("myfile");
  if(!out) cout<<"\n Cannot open file";
  else
  { out<<" this is my first program – file handling";
    out.close( ); }
}
```

Creating sequential file

- Open the file
- Read/write sequentially
- Close the file

Example

```
#include<iostream>
#include<fstream>
using namespace std;
void main( )
{ ofstream out("inventory");
  if(!out) cout<<"\n Cannot open file";
  else
  { out<<" Radios " <<40<<endl ;
    out<<" TV " <<20<<endl ;
    out<<" iPods " <<100<<endl ;
    out.close( ); }
}
```

Reading from file

```
#include<iostream>
#include<fstream>
using namespace std;
void main( )
{ ifstream in("inventory");
  if(!in) cout<<"\n Cannot open file";
  else
  { char item[20]; float n;
    in>>item>>cost ; cout<<"ITEM : "<<item<<" Number : "<<n<<endl ;
    in>>item>>cost ; cout<<"ITEM : "<<item<<" Number : "<<n<<endl ;
    in>>item>>cost ; cout<<"ITEM : "<<item<<" Number : "<<n<<endl ;
    in.close( ); }
}
```