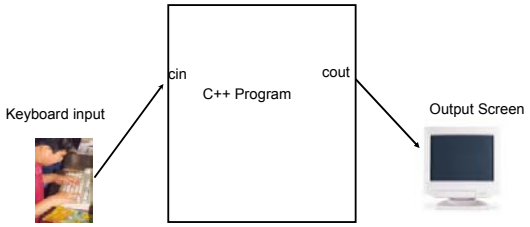
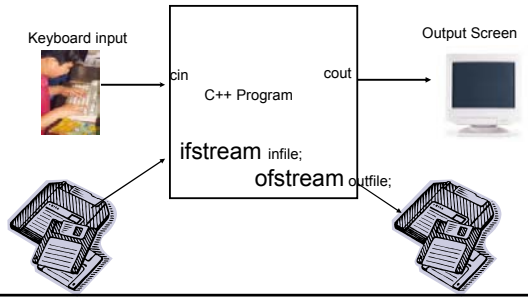
Programming with Data Files

Chapter 4

Standard Input Output



File Input / Output features



Programmer Defined File Streams

- To define your own file streams
  - for input use ifstream class
  - for output use ofstream class
- ifstream and ofstream are defined in package fstream

File Operations

- Open, close, << and >> operators
- eof() operation on an input file object returns a true or false (Boolean)
- get() reads a single character from an input file and put(char) writes a single character into an output file.

Defining File Streams

1. Include **fstream**
2. declare file stream variable (object)
   1. **ifstream fin;**
   2. **ofstream fout;**
3. use open() to initialize file stream variable
4. use input file stream variable as you would use **cin** and use output file stream variable as you would use **cout**
5. use close() to close the file when finished with it

```
#include <fstream>
#include <cmath>
using namespace std;
int main()
{  //Create three columns of data for plotting.
   double x;
   ifstream xdata;      //declare input stream
   ofstream plotdata;          //declare output stream
   xdata.open("data1");    //xdata uses file data1
   if( xdata.fail() )          //check for error
   {  cout << "error opening input file" << endl;
      return 0;
   }  //end if fail
   plotdata.open("plot1");//plotdata uses file plot1
   xdata >> x;                 //input x from file
   while(!xdata.eof())         //while not end of file
   {  if( x>0 )                //write to plot file
         plotdata<<x<<" "<<exp(x)<<" "<<log(x)<<endl;
      xdata >> x;      //get next data point
   }  //end while
     xdata.close();
     plotdata.close();
      return 0;
}  //end main
```


Reading Data Files

- Specified number of records
  - for loop
- Trailer or Sentinel Signal
  - while loop
- Data records only (no specified number of records, no sentinel value)
  - while loop

Example - Specified Number of Records

```
…
int main()
{
  fstream fin("data");
  double x;
  int num_data_points;
  …
  fin >> num_data_points;
  for(int i=0; i<num_data_points; i++)
  {
    fin >> x;
    …
  }
  …
```


Example - Sentinel Signal

```
const double sentinelValue = -99;
…
int main()
{
  fstream fin("data");
  double x;
  …
  fin >> x;
  while(x != sentinelValue)
  {
    …
    fin >> x;
  }
…
```


Example - No Specified Number of Records, no Sentinel Signal

```
…
int main()
{
  fstream fin("data");
  double x;
  …
  fin >> x;
  while( !fin.eof() )
  {
    …
    fin >> x;
  }
  …
```


Summary

- We looked at
  - Standard Input and Output
  - File Input and output features
  - Programmer Defined Files
  - File operations
  - Reading multiple data sets from input file