

```
1 //-----
2 // File p02.cpp
3 //-----
4 // Project p02 is comprised of the following files:
5 // File      Description
6 //
7 // p02.cpp    Processes command line arguments and directs the process
8 //            the Cartesian product.
9 //
10 // List02.h   Defines class List - a list of strings
11 //
12 // List02.cpp Implements class List
13 //
14 // p02make    Creates executable file p02 from files p02.cpp,
15 //            List02.cpp, and List02.h
16 //-----
17 //Author:     Ms. Petunia Perfect
18 //Student ID: *00000000
19 //Course:     CMSC 2123 - Discrete Structures
20 //CRN:        11786, Autumn, 2015
21 //Project:    p02
22 //Due:        September 18, 2015
23 //Account:    tt000
24 //-----
25 //Copyright September, 2015 by Thomas R. Turner.
26 //Do not reproduce without permission from Thomas R. Turner.
27 //-----
28 //C++ standard include files.
29 //-----
30 #include <cstdlib>
31 #include <cstring>
32 #include <iostream>
33 #include <string>
34 #include <fstream>
35 #include <iomanip>
36 using namespace std;
37 //-----
38 //Application include files
39 //-----
40 #include "List02.h"
41 //-----
42 //Function CartesianMgr reads the input file stream, produces
43 //the Cartesian product, and writes it to the output file stream.
44 //-----
45 void CartesianMgr(istream& i1,istream& i2,ostream& o)
46 {
47     List L1;
48     L1.Scan(i1);
49     L1.Print(o,"L1");
50 }
51 //-----
52 //Exception CommandLineException is thrown when more than strings
53 //appear on the command line
54 //-----
55 struct CommandLineException {
56     CommandLineException(int m,int a)
57     {
58         cout << endl;
59         cout << "Too many file names on the command line.";
60         cout << endl;
```

```

59         cout << m << " file names are permitted.";
60         cout << endl;
61         cout << a << " file names were entered.";
62         cout << endl;
63     }
64 };
65 //-----
66 //Exception FileException is thrown when a file whose name is
67 //given on the command line cannot be opened.
68 //-----
69 struct FileException {
70     FileException(char* fn)
71     {
72         cout << endl;
73         cout << "File " << fn << " could not be opened.";
74         cout << endl;
75     }
76 };
77 //-----
78 //Function main processes command line arguments
79 //-----
80 int main(int argc, char* argv[])
81 {
82     try {
83         char ifn1[255];           //Input File Name 1
84         char ifn2[255];           //Input File Name 2
85         char ofn[255];            //Output File Name
86         switch (argc) {
87             case 1:                //Prompt for all file names
88                 cout << "Enter input file name 1. ";
89                 cin >> ifn1;
90                 cout << "Enter input file name 2. ";
91                 cin >> ifn2;
92                 cout << "Enter the output file name. ";
93                 cin >> ofn;
94                 break;
95             case 2:                //Copy the input file name from argv
96                                     //Prompt for the other input file name and the o
97                                     utput file name
98                 strcpy(ifn1,argv[1]);
99                 cout << "Enter input file name 2. ";
100                 cin >> ifn2;
101                 cout << "Enter the output file name. ";
102                 cin >> ofn;
103                 break;
104             case 3:                //Copy both input file names from argv
105                 strcpy(ifn1,argv[1]);
106                 strcpy(ifn2,argv[2]);
107                 cout << "Enter the output file name. ";
108                 cin >> ofn;
109                 break;
110             case 4:                //Copy all file names from argv
111                 strcpy(ifn1,argv[1]);
112                 strcpy(ifn2,argv[2]);
113                 strcpy(ofn,argv[3]);
114                 break;
115             default:
116                 throw CommandLineException(3,argc-1);
117         }
118         ifstream i1(ifn1); if (!i1) throw FileException(ifn1);
119         ifstream i2(ifn2); if (!i2) throw FileException(ifn2);
120         ofstream o(ofn); if (!o) throw FileException(ofn);

```

```
119         CartesianMgr(i1,i2,o);
120         o.close();
121         i1.close();
122         i2.close();
123
124     } catch (...) {
125         cout << endl;
126         cout << "Program terminated.";
127         cout << endl;
128         cout << "I won't be back!";
129         cout << endl;
130         exit(EXIT_FAILURE);
131     }
132     return 0;
133 }
```