7 Text I/O

REVIEW QUESTIONS

1. Which of the following is considered auxiliary storage?

e. both disk and tape

3. The four standard streams in C++ are

b. cin, cout, cerr, and clog

5. To open a file for output, we need to use an object of type _____.

d. ofstream

7. The showbase flag prints the data in octal or hexadecimal.

a. true

9. The default fill character in C++ is _____.

a. a space

11. All systems provide manipulators for all flags and control variables.

b. false

13. Find the error (in any) in each of the following statements.

a. cin.get (c)

b. cin.get (c)

c. No errors

d. cout.put (c)

EXERCISES

15. The values are:

i1 = 14 i2 = 23 f1 = 76.0 c1 = 'C' c2 = 'D' c3 = '\n'

17. The values are:

i1 = 45 i2 = 123 i3 = 34 c1 = 'c' c2 = '1' c3 = 'd' and ".7\n" remains in the input stream.

PROBLEMS

19.

```
/* ============ append_file ==========
   This function appends one file to the another.
      Pre  Files exist
      Post File 2 appended to file 1
           and written to file 3
*/
int append_file ()
{
   ifstream fs1;
   fs1.open("file1.dat");
   if (!fs1)
      {
       cout << "\nError opening file1.dat\n";
       return (1);
      } // if

   ifstream fs2;
   fs2.open("file2.dat");
   if (!fs2)
      {
       cout << "\nError opening file2.dat\n";
       return (2);
      } // if

   ofstream fs3;
   fs3.open("file3.dat");
   if (!fs3)
      {
       cout << "\nError opening file3.dat\n";
       return (3);
      } // if

   char c;
   while (fs1.get (c))
       fs3.put (c);

   while (fs2.get (c))
       fs3.put (c);

   fs1.close();
   fs2.close();
   fs3.close();
   return 0;
}  // append_file
```

21.

```
/* ============ format_line ============
   Reformat file with 60 characters per line.
      Pre  File 1 exists
      Post Reformatted file created
*/
int format_line ()
{
   ifstream fs1;
   fs1.open ("FILE2.DAT");
   if (!fs1)
      {
       cout << "\nError 1: opening file for
reading.\n";
```

```
        return (1);
      } // if

   ofstream fs2;
   fs2.open ("FILE2.DAT");
   if (!fs2)
      {
       cout << "\nError 2: opening file for
writing.\n";
       return (2);
      } // if

   char c;
   int  count = 0;
   while (fs1.get (c))
      {
       if (count < 60)
          {
           if (c != '\n')
              fs2.put (c);
           else
              fs2.put (' ');
           count++;
           } // count < 60
       else
           {
            fs2.put ('\n');

            if (c != '\n')
               {
                fs2.put (c);
                count = 1;
               } // if
            else
                count = 0;
           } // else
      }  // while !EOF

   if (count)
      fs2.put ('\n');

   fs1.close();
   fs2.close();
   return 0;
}  // format_line
```

23.

```
/* ========== del_last_line ==================
   This function deletes the last line of any file.
      Pre  File 1 exists
      Post Last line of file deleted
*/
int del_last_line ()
{
   ifstream fs1;
   fs1.open ("FILE1.DAT");
   if (!fs1)
      {
       cout << "\nError 1L opening file for
reading.\n";
```

```
         return (1);
        } // if

    ofstream fs2;
    fs2.open ("FILE8.DAT");
    if (!fs2)
        {
         cout << "\nError 2: opening file\n";
         return (2);
        } // if

    char c;
    int  line_cnt = 0;
    while (fs1.get (c))
        {
         if (c  ==  '\n')
            line_cnt++;
        }
    fs1.close();

    fs1.open ("FILE1.DAT");
    if (!fs1)
        {
         cout << "\nError 3: opening file\n";
         return (3);
        } // if

    line_cnt--;
    while (line_cnt)
        {
         fs1.get(c);
         if (c == '\n')
            line_cnt--;
         fs2.put (c);
        } // while
    fs1.close();
    fs2.close();

    return 0;
}   // del_last_line
```

25.

```
/* This program prints itself.
      Written by:
      Date:
*/
#include <iostream>
#include <fstream>
using namespace std;

int main ()
{
    cout << "*** start of print program ***\n\n";

    ifstream fs1;
    fs1.open ("prb0735.cpp");
    if (!fs1)
        {
         cout << "\nError 1: opening program.\n";
         return (1);
```

```
    } // if

    char     c;
    while (fs1.get (c))
        cout << c;
    fs1.close();
    cout << "\n ***  end of print ***\n\n";
    return 0;
}   // main
```

27.

```
/*  Parse words and write them as separates lines
    to a new file.
       Written by:
       Date:
*/
#include <iostream>
#include <fstream>
using namespace std;

int main ()
{
    cout << "\n *** start of program ***\n\n";

    ifstream fs1;
    fs1.open ("FILE1.DAT");
    if (!fs1)
       {
        cout << "\nError 1: opening file for
reading.\n";
        return (1);
       } // if

    ofstream fs2;
    fs2.open ("FILE13.DAT");
    if (!fs2)
       {
        cout << "\nError 2: opening file for
writing.\n";
        return (2);
       } // if

    char cur;
    bool flag = false;
    while (fs1.get (cur))
       {
        if (cur == ' ' || cur == '\n' || cur == '\t')
           {
            if (flag == true)
               {
                fs2.put ('\n');
                flag = false;
               } // if flag
           } // if whitespace
        else
           {
            fs2.put (cur);
            if (flag != true)
               flag = true;
           } // else
```

```
      } // while

   fs1.close ();
   fs2.close ();
   cout << "\n ***  end of program ***\n\n";
   return 0;
}  // main
```

29.

```
/* This program inserts a blank line after the seventh
   line in the file.
      Written by:
      Date:
*/
#include <iostream>
#include <fstream>
using namespace std;

int main ()
{
   cout << "\n *** start of program ***\n\n";

   ifstream   fs1;
   fs1.open ("FILE1.DAT");
   if (!fs1)
      {
       cout << "\nError 1: opening file\n";
       return (1);
      } // if

   ofstream   fs2;
   fs2.open ("FILE15.DAT");
   if (!fs2)
       {
       cout << "\nError 2: opening file\n";
       return (2);
      } // if

   char c;
   int  line_cnt = 0;
   while (fs1.get (c))
      {
       if (c == '\n')
          line_cnt++;
       if (c == '\n' && line_cnt == 7)
          fs2.put ('\n');
       fs2.put (c);
      } // while

   fs1.close ();
   fs2.close ();
   cout << "\n ***  end  of program ***\n\n";
   return 0;
}  // main
```

31.

```
/* Insert a blank line after each line in a file.
      Written by:
      Date:
*/
```

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

int main ()
{
   cout << "\n *** start of program ***\n\n";

   ifstream   fs1;
   fs1.open ("FILE1.DAT");
   if (!fs1)
      {
       cout << "\nError 1: opening file\n";
       return (1);
      } // if

   ofstream   fs2;
   fs2.open ("FILE17.DAT");
   if (!fs2)
      {
       cout << "\nError 2: opening file\n";
       return (2);
      } // if

   char  ch;
   while (fs1.get (ch))
      {
       if (ch == '\n')
          fs2.put ('\n');
       fs2.put (ch);
      } // while

   fs1.close ();
   fs2.close ();
   cout << "\n ***  end  of program ***\n\n";
   return 0;
}  // main
```

33.

```
/* This program copies a file, deleting the first two
   characters of each line.
      Written by:
      Date:
*/
#include <iostream>
#include <fstream>
using namespace std;

int main ()
{
   cout << "\n *** start of program ***\n\n";

   ifstream   fs1;
   fs1.open ("FILE1.DAT");
   if (!fs1)
      {
       cout <<
          "\nError 1: opening file for reading.\n";
       return (1);
      } // if
```

```
   ofstream   fs2;
   fs2.open ("FILE19.DAT");
   if (!fs2)
      {

       cout <<
          "\nError 2: opening file for writing.\n";
       return (2);
      } // if

   char       c;
   int        char_cnt = 0;
   while (fs1.get (c))
      {
       if (c != '\n')
          char_cnt++;
       else
          char_cnt = 0;

       if (char_cnt != 1 && char_cnt != 2)
          fs2.put (c);
      } // while

   fs1.close ();
   fs2.close ();
   cout << "\n ***  end  of program ***\n\n";
   return 0;
}  // main
```