

## 1 Review of File Concepts

- Files store information for later use.
- Naming convention dependent upon Operating System.
- Typical names

prog1.cpp	C++ source file
prog1.h	C++ header file
test.in	input file
test.out	output file
prog1.exe	Executable file (application)
prog1.obj	object file
report.doc	Word processor document
myPage.html	HTML document

## 2 Files

- Files can be treated as stream objects.
- Basic I/O stream types
  - ifstream — input file stream
  - ofstream — output file stream
  - The header files `fstream.h` and `iomanip.h` may also need to be included—system/compiler dependent
- Basic operations
  - Member functions (partial list)
    - `fill( char c )` sets the fill character to `c`
    - `setf( long f )` set flags in `f` to 1
    - `unsetf( long f )` set flags in `f` to 0
  - iostream manipulators (partial list)

```
endl    output newline and flush the stream
ends    output null character '\0'
flush   flush the stream
dec     display numeric values using decimal
hex     display numeric values using hexadecimal
oct     display numeric values using octal
ws      skip over leading white space
```

– Output manipulators (partial list)

```
setw( int w )           set field width to w—not persistent!
setprecision( int d )   set accuracy to d
scientific( )           use scientific format d.dddEdd
fixed( )                use floating-point format dddd.ddd
setbase( int b )        set the numeric base to b
setfill( char c )       sets the fill character to c
setiosflags( long f )   set flags in f to 0
resetiosflags( long f ) set flags in f to 1
```

– iosflags ios:: (partial list)

```
ios::left    left justify
ios::right   right justify
ios::showpoint show decimal point
ios::skipws  skip white space
```

ios will become ios\_base in the future

:: is the scope resolution operator

## 2.1 File operations

- Basic operations
  - open()
  - close()
  - is\_open()
- open modes (compiler dependent)

<code>in</code>	open for reading
<code>out</code>	open for writing
<code>app</code>	append
<code>trunc</code>	truncate file to zero-length
<code>ate</code>	open and move to end of file (pronounced “at end”)
<code>binary</code>	I/O in binary mode (rather than text)

## 2.2 Opening an output file

- Open Operation

```
ofstream fOut;  
fOut.open( "test.out", ios::out );
```

- Declare and Open

```
ofstream fOut( "test.out", ios::out );
```

- Verify opening

```
if( !fOut )  
{  
    cout << "Error opening \"test.out\" << endl;  
    exit( -1 );  
}
```

- close

```
fOut.close();
```

## 2.3 Opening an input file

- Open Operation

```
ifstream fIn;  
fIn.open( "test.in", ios::in );
```

- Declare and Open

```
ifstream fIn( "test.in", ios::in );
```

- Verify opening

```
if( !fIn )  
{  
    cout << "Error opening \"test.in\" \" << endl;  
    exit( -1 );  
}
```

- close

```
fIn.close();
```

## 2.4 *sine* Table: Take 4

```
// sTable4.cpp

#include <iostream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <math.h>

main()
{
    ofstream fOut( "sTable.out", ios::out );
    if( !fOut )          // verify file was opened
    {
        cerr << "Unable to open output file:  sTable.out" << endl;
        exit( -1 );
    }
    // constants
    const double PI = 3.14159;
    const double DEG_TO_RAD = PI/180.0;

    double xStart = 0.0;    // starting value (degrees)
    double xEnd   = 30.0;   // final value
    double xInc   = 5.0;    // increment

    fOut << setw(5) << "x" << setw(10) << "sin(x)" << endl;

    for( double x = xStart ; x <= xEnd ; x += xInc )
    {
        double xRad = x * DEG_TO_RAD; // convert to radians
        double s = sin(xRad);

        fOut << setw(7) << x << setw(10) << s << endl;
    }

    fOut.close();
}
```

Output:

x	sin(x)
0	0
5	0.0871557
10	0.173648
15	0.258819
20	0.34202
25	0.422618
30	0.5

Limit number of digits displayed using `setprecision`:

```
fOut << setw(5) << "x" << setw(10) << "sin(x)" << endl;

fOut << setprecision(3);

for( double x = xStart ; x <= xEnd ; x += xInc )
    ....
```

Output:

x	sin(x)
0	0
5	0.0872
10	0.174
15	0.259
20	0.342
25	0.423
30	0.5

Is this what we expected?

noindent Display decimal point using `setiosflags`:

```
fOut << setw(5) << "x" << setw(10) << "sin(x)" << endl;

fOut << setprecision(3);
fOut << setiosflags( ios::fixed | ios::showpoint );

for( double x = xStart ; x <= xEnd ; x += xInc )
    ....
```

Output:

x	sin(x)
0.000	0.000
5.000	0.087
10.000	0.174
15.000	0.259
20.000	0.342
25.000	0.423
30.000	0.500



```
// quadFile1.cpp

#include <iostream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    // Open input file
    ifstream fIn( "quad.in", ios::in );
    if( !fIn )           // verify file was opened
    {
        cerr << "Unable to open output file:  quad.in" << endl;
        exit( -1 );
    }

    char  ans;           // answer
    double a, b, c;     // coefficients

    do
    {
        // Get coefficients from user
        fIn >> a >> b >> c;

        // Echo values
        cout << "The polynomial coefficients are: " << endl;
        cout << "a:  " << a << endl;
        cout << "b:  " << b << endl;
        cout << "c:  " << c << endl;

        double discr = b*b - 4.0*a*c;
        if( discr <= 0.0 )
        {
            cout << "Unable to solve quadratic equation:" << endl;
            cout << "\tDiscriminant is less than or equal zero" << endl;
        }
    }
}
```

```
else
{
    discr = sqrt(discr);
    double denom = 2.0 * a;
    double x1 = (-b + discr) / denom;
    double x2 = (-b - discr) / denom;

    cout << "x1: " << x1 << endl;
    cout << "x2: " << x2 << endl;
}

do
{
    // prompt
    cout << "Continue (y/n)? " << flush;
    // read answer
    fIn >> ans;

    cout << ans << endl;
} while ( (ans != 'y') && (ans != 'n') );
} while ( (ans == 'y') );

// Close input file
fIn.close();

cout << " Done!" << endl;

return 0;
}
```

Input file: quad.in

```
2.0 3.0 5.0
y
2.0 5.0 3.0
y
2.0 6.0 2.0
n
```

Output (written to screen):

```
The polynomial coefficients are:
a: 2
b: 3
c: 5
Unable to solve quadratic equation:
    Discriminant is less than or equal zero
Continue (y/n)? y
The polynomial coefficients are:
a: 2
b: 5
c: 3
x1: -1
x2: -1.5
Continue (y/n)? y
The polynomial coefficients are:
a: 2
b: 6
c: 2
x1: -0.381966
x2: -2.61803
Continue (y/n)? n
Done!
```

```
// quadFile2.cpp

#include <iostream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    // Open input and output files
    ifstream fIn( "quad.in", ios::in );
    if( !fIn )           // verify file was opened
    {
        cerr << "Unable to open output file:  quad.in" << endl;
        exit( -1 );
    }

    ofstream fOut( "quad.out", ios::out );
    if( !fOut )         // verify file was opened
    {
        cerr << "Unable to open output file:  quad.out" << endl;
        exit( -1 );
    }

    char  ans;           // answer
    double a, b, c;     // coefficients

    do
    {
        // Get coefficients from user
        fIn >> a >> b >> c;

        // Echo values
        fOut << "The polynomial coefficients are: " << endl;
        fOut << "a:  " << a << endl;
        fOut << "b:  " << b << endl;
        fOut << "c:  " << c << endl;
    }
}
```

```
double discr = b*b - 4.0*a*c;
if( discr <= 0.0 )
{
    fOut << "Unable to solve quadratic equation:" << endl;
    fOut << "\tDiscriminant is less than or equal zero" << endl;
}
else
{
    discr = sqrt(discr);
    double denom = 2.0 * a;
    double x1 = (-b + discr) / denom;
    double x2 = (-b - discr) / denom;

    fOut << "x1: " << x1 << endl;
    fOut << "x2: " << x2 << endl;
}

do
{
    // prompt
    fOut << "Continue (y/n)? " << flush;
    // read answer
    fIn >> ans;

    fOut << ans << endl;
} while ( (ans != 'y') && (ans != 'n') );
} while ( (ans == 'y') );

// Close files
fIn.close();
fOut.close();

cout << " Done!" << endl;

return 0;
}
```

Input file: quad.in

```
2.0 3.0 5.0
y
2.0 5.0 3.0
y
2.0 6.0 2.0
n
```

Output (stored in quad.out):

```
The polynomial coefficients are:
a: 2
b: 3
c: 5
Unable to solve quadratic equation:
Discriminant is less than or equal zero
Continue (y/n)? y
The polynomial coefficients are:
a: 2
b: 5
c: 3
x1: -1
x2: -1.5
Continue (y/n)? y
The polynomial coefficients are:
a: 2
b: 6
c: 2
x1: -0.381966
x2: -2.61803
Continue (y/n)? n
```