

<<<< USERNAME:tt FILENAME: masfiles.cpp >>>>

```
//-----
//File masFiles.cpp opens and closes files specific to the MARIE
//assembler.
//-----
//Author: Thomas R. Turner
//E-Mail: trturner@uco.edu
//Date: August, 2016
//-----
//Copyright August, 2016 by Thomas R. Turner
//Do not reproduce without permission from Thomas R. Turner
//-----
//C++ Standard include files
//-----
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstdio>
#include <string>
using namespace std;
//-----
//Application include files
//-----
#include "masfiles.h"
//-----
//-----
//BadSuffixException
//-----
struct BadSuffixException {
    BadSuffixException(char* fn)
    {   cout << endl;
        cout << "Input file \"" << fn << "\" does not have a .mas suffix.";
    }
};
//-----
//-----
class FileNameSuffix {
    char* prefix;
public:
    FileNameSuffix(char* fn)
    {   char* p=strstr(fn, ".mas");
        if (!p) throw BadSuffixException(fn);
        int n=p-fn;
        if (n+4!=strlen(fn)) throw BadSuffixException(fn);
        prefix=new char[strlen(fn)+1];
        strncpy(prefix, fn, n);
        prefix[n]=0;
    }
    ~FileNameSuffix(){if (prefix) delete[] prefix;}
    void Suffix(char* fn, const char* suffix)
    {   strcpy(fn, prefix);
        strcat(fn, suffix);
    }
};
//-----
//FileException
//-----
struct FileException {
    FileException(const char* fn)
    {   cout << endl;
        cout << "File " << fn << " could not be opened.";
        cout << endl;
    }
};
};
```

<<<< USERNAME:tt FILENAME: masfiles.cpp >>>>

```
//Class Files
//-----
Files::Files() {}
Files::~Files()
{
    if (ifn) delete[] ifn;
    if (tfn) delete[] tfn;
    if (efn) delete[] efn;
}
void Files::Open(char* fn)
{
    //-----
    //Allocate storage for file names
    //-----
    ifn=new char[strlen(fn)+1];    //Allocate storage for the input file name
    strcpy(ifn,fn);                //Copy the file name
    tfn=new char[strlen(ifn)+1];    //Allocate storage for the trace file name
    efn=new char[strlen(ifn)+1];    //Allocate storage for the executable file nam
e
    //-----
    //Open the input source file
    //-----
    i=fopen(ifn,"r");
    if (!i) throw FileException(ifn);

    //-----
    //Create the trace file name and open the trace file
    //-----
    FileNameSuffix FS(ifn);        //Find the prefix of the input file name
    FS.Suffix(tfn,".trc");          //Create the trace file name
    tfs.open(tfn); if (!tfs) throw FileException(tfn);
    if (!tfs) throw FileException(tfn);

    //-----
    //Create the executable file name and open the executable file
    //-----
    FS.Suffix(efn,".mex");
    e=fopen(efn,"wb");
    if (!e) throw FileException(efn);
}
void Files::Close()
{
    if (i) fclose(i);
    if (tfs) tfs.close();
    if (e) fclose(e);
}
```