```
::::::::::::::
Figures.cpp
::::::::::::::
# include <iostream>
# include <iomanip>
# include <cstdlib>
# include <cmath>
# include <fstream>
# include <vector>
# include <algorithm>
# include <cstring>
# include <string>

# include "Figures.hpp"


using namespace std;


//****************************************************************************
//   Purpose:
//
//     Figures Class Implementation Files
//
//  Discussion:
//
//
//  Licensing:
//
//    This code is distributed under the GNU LGPL license.
//
//  Modified:
//
//    2013.05.11
//
//  Author:
//
//    Young Won Lim
//
//  Parameters:
//
//----------------------------------------------------------------------------
//
// Figures::Figures()
// Figures::insert_fig_basic_files(ofstream& myout)
// Figures::insert_fig_tscale_files(ofstream& myout)
// Figures::insert_fig_uscale_files(ofstream& myout)
// Figures::make_latex_file(int mode)
// Figures::make_figures(int mode)
//----------------------------------------------------------------------------
```

```
//--------------------------------------------------------------------------
//  Class Figures' Member Functions
//--------------------------------------------------------------------------
Figures::Figures()
{
   epsList.clear();

}



//--------------------------------------------------------------------------
// include basic figure eps files
//--------------------------------------------------------------------------
void Figures::insert_fig_file_list(ofstream& myout) {

  list<string>::iterator I;
  int count=0;


  myout << "\\begin{lstlisting}" << endl;
  for (I = epsList.begin(); I != epsList.end(); ++I) {
    myout << *I << endl;
  }
  myout << "\\end{lstlisting}" << endl;


  myout << "\\newpage" << endl;

  myout << "\\begin{figure}[h!]" << endl;
  myout << "\\begin{center}" << endl;
  for (I = epsList.begin(); I != epsList.end(); ++I) {
    // myout << "\\includegraphics[scale=0.5]{./";
    myout << "\\includegraphics[width=0.48\\textwidth]{./";
    myout << *I << "}" << endl;

    count++;
    // if (count == epsList.size()/2) {
    if (count %8  == 0) {
      myout << "\\end{center}" << endl;
      myout << "\\end{figure}" << endl;
      myout << "\\newpage" << endl;
      myout << "\\begin{figure}[h!]" << endl;
      myout << "\\begin{center}" << endl;
    }

  }
```

```
  myout << "\\end{center}" << endl;
  myout << "\\end{figure}" << endl;

}


//--------------------------------------------------------------------------
// Making tex output files
//--------------------------------------------------------------------------
void Figures::make_latex_file() {
  ofstream myout;



  // writing gnuplot commands
  myout.open(fname);


  myout << "\\documentclass[12pt]{article}" << endl;
  myout << "" << endl;
  myout << "\\usepackage{amsmath}    % need for subequations" << endl;
  myout << "\\usepackage{graphicx}   % need for figures" << endl;
  myout << "\\usepackage{verbatim}   % useful for program listings" << endl;
  myout << "\\usepackage{color}      % use if color is used in text" << endl;
  myout << "%\\usepackage{subfloat}  % use for side-by-side figures" << endl;
  myout << "%\\usepackage{hyperref}   % use for hypertext links" << endl;

  myout << "\\usepackage{listings}"  << endl;

  myout << "" << endl;
  myout << "" << endl;
  myout << "\\setlength{\\baselineskip}{16.0pt}    % 16 pt usual spacing between lines" << endl;
  myout << "" << endl;
  myout << "%\\setlength{\\parskip}{3pt plus 2pt}" << endl;
  myout << "%\\setlength{\\parindent}{20pt}" << endl;
  myout << "%\\setlength{\\oddsidemargin}{0.5cm}" << endl;
  myout << "%\\setlength{\\evensidemargin}{0.5cm}" << endl;
  myout << "%\\setlength{\\marginparsep}{0.75cm}" << endl;
  myout << "%\\setlength{\\marginparwidth}{2.5cm}" << endl;
  myout << "%\\setlength{\\marginparpush}{1.0cm}" << endl;
  myout << "%\\setlength{\\textwidth}{150mm}" << endl;
  myout << "\\addtolength{\\oddsidemargin}{-2.5cm}" << endl;
  myout << "\\addtolength{\\evensidemargin}{-2.5cm}" << endl;
  myout << "\\addtolength{\\marginparwidth}{-3.0cm}" << endl;
  myout << "\\addtolength{\\textwidth}{+4.0cm}" << endl;
  myout << "" << endl;
  myout << "\\begin{document}" << endl;
  myout << "" << endl;
  myout << "\\begin{center}" << endl;
```

```
  myout << "{\\large " << title << "} \\\\" << endl;
  myout << "\\today" << endl;
  myout << "\\end{center}" << endl;
  myout << "" << endl;
  myout << "" << endl;
  myout << "" << endl;

  //................................
  insert_fig_file_list(myout);
  //................................

  myout << "" << endl;
  myout << "" << endl;
  myout << "" << endl;
  myout << "\\end{document}" << endl;

  myout.close();


}


//-------------------------------------------------------------------------
// Determine the kinds of tex output files to be written
//-------------------------------------------------------------------------
void Figures::make_figures(int mode, list<string>& epsL1, list<string>& epsL2) {

  list<string>::iterator I;

  epsList.clear();

  for (I = epsL1.begin(); I != epsL1.end(); ++I) {
    epsList.push_back(* I);
    cout << * I << endl;
  }

  for (I = epsL2.begin(); I != epsL2.end(); ++I) {
    epsList.push_back(* I);
    cout << * I << endl;
  }

  for (I = epsList.begin(); I != epsList.end(); ++I) {
    cout << * I << endl;
  }


   cout << "mode= " << mode << endl;

   if (mode & 1) {
```

```
      strcpy(fname, "fig_basic.tex");
      strcpy(title, "Basic Figures");

      make_latex_file();
      system("latex fig_basic.tex");
      cout << "end of latex \n";
      system("dvipdf fig_basic.dvi");
      cout << "end of dvipdf \n";
    }

    if (mode & 2) {
      strcpy(fname, "fig_tscale.tex");
      strcpy(title, "TScale Figures");

      make_latex_file();
      system("latex fig_tscale.tex");
      system("dvipdf fig_tscale.dvi");
    }

    if (mode & 4) {
      strcpy(fname, "fig_uscale.tex");
      strcpy(title, "UScale Figures");

      make_latex_file();
      system("latex fig_uscale.tex");
      system("dvipdf fig_uscale.dvi");
    }

}

::::::::::::::
Figures.hpp
::::::::::::::
# include <iostream>
# include <iomanip>
# include <fstream>
# include <string>
// # include <cstdlib>
// # include <cmath>
# include <vector>
# include <algorithm>
# include <map>
# include <list>

using namespace std;


//------------------------------------------------------------------------
//   Purpose:
```

```
//
//     Class Figures Interface Files
//
//  Discussion:
//
//
//  Licensing:
//
//    This code is distributed under the GNU LGPL license.
//
//  Modified:
//
//    2013.05.11
//
//  Author:
//
//    Young Won Lim
//
//  Parameters:
//
//------------------------------------------------------------------------

class Figures
{

  public:

  Figures();

  char fname[200];
  char title[200];

  void insert_fig_file_list(ofstream& myout);
  void make_latex_file();

  void make_figures(int mode, list<string>& epsL1, list<string>& epsL2);

  list<string> epsList;

};
```