Comp151 Lab
Documentation using Doxygen

Supplementary Notes
By Adam

Information in this slide is extracted from Doxygen homepage:
http://www.stack.nl/~dimitri/doxygen/
and Javadoc reference:
http://java.sun.com/j2se/javadoc/writingdoccomments/

Introduction

- Doxygen is a documentation system for C++, C, Java, Objective-C, Python, IDL (Corba and Microsoft flavors) and to some extent PHP, C#, and D.
- It can generate documentations in HTML or LATEX format from a set of documented source files. There is also support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages.
- It is distributed under GNU General Public License. You are permitted to use, copy, modify, and distribute this software and its documentation under the terms of this license.

Documenting the source code

Hello.hpp

```
/** \file hello.hpp
 *  Descriptions about this file.
 */
#include <iostream>
…
/**
 *  \class hello
 *  \author  Helloman
 *  \brief  This is a brief description
 *
 *  Here is some detail descriptions.
 */
class hello {
public:
    int index; //!< This is a data member
    …
    /**
     *  \brief  This is a brief description for print()
     */
    void print (String txt //!< This is a parameter
          );
};
```

- Documentations are written inside comment blocks with special indicators, e.g. /** and */ or //!<.
- They are usually placed before definition or declarations, although they can be put in other places.
- Doxygen will collect them and generate appropriate documentations.

Documentation for class

```
/**
 * \class CosFunc.
 * \brief A class derived from Func to implement cosine function.
 */
class CosFunc: public Func {…};
```

- \class and \brief are special commands. Special commands can start with “\” or “@”.
- They tell Doxygen how to handle a piece of information.

\class

- \class <name>
  - Indicates that a comment block contains documentation for a class.
  - You can omit this one if the block is right before the definition of a class.
- In writing documentation, you can omit the subject and simply state the object. That will make it easier to read, e.g.
  - “This class is the base class of all function classes.” We can use “The base class of all function classes.” instead.
- The class inheritance hierarchy diagram will be automatically generated by Doxygen.

\brief

- \brief { brief description }
  - Starts a paragraph that serves as a brief description. It ends when a blank line or another sectioning command is encountered.
  - If multiple \brief commands are present they will be joined.
  - Detailed description can be put after \brief separated by a blank line.
- It can be used in many kinds of documentation blocks, e.g. blocks for classes, functions or files
- You can use phrases instead of complete sentences in writing \brief, e.g. “To create a cosine function.”

\brief (con’t)

- Question:

  If I write documentation blocks for an element in both header and source file, which one will be used?

- For \brief,
  - only the one before the *declaration* will be used. (usually in the header file)

- For detailed description,
  - the one before *definition* is preferred and the other one will be ignored. (usually in the source file)

lab07: CosFunc Class Reference - Microsoft Internet Explorer

File Edit View Favorites Tools Help

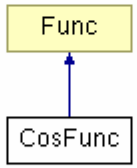
Main Page | Namespace List | Class Hierarchy | Class List | File List | Class Members | File Members | Related Pages

CosFunc Class Reference

A class derived from **Func** to implement cosine function. More...

```
#include <tri_func.hpp>
```

Inheritance diagram for CosFunc:



Class Inheritance Hierarchy Diagram

List of all members.

Public Member Functions

	CosFunc (double _scale=1.00) *To create a cosine function.*
double	**EvaluateAt** (double d) const *Evaluate the function at d.*
double	**DerivativeAt** (double d) const *Calculate the derivative value of a function at d.*

Sentences after \brief will be displayed here.

My Computer

Documentation for function

```
/**
 * \brief Evaluate the function at d.
 *
 * \return a function value
 */
double EvaluateAt(double d   //!< value in the domain
                  ) const;
```

- \return { description of the return value }
  - Starts a return value description for a function.
  - Like \brief, the \return description ends when a blank line or some other sectioning command is encountered.

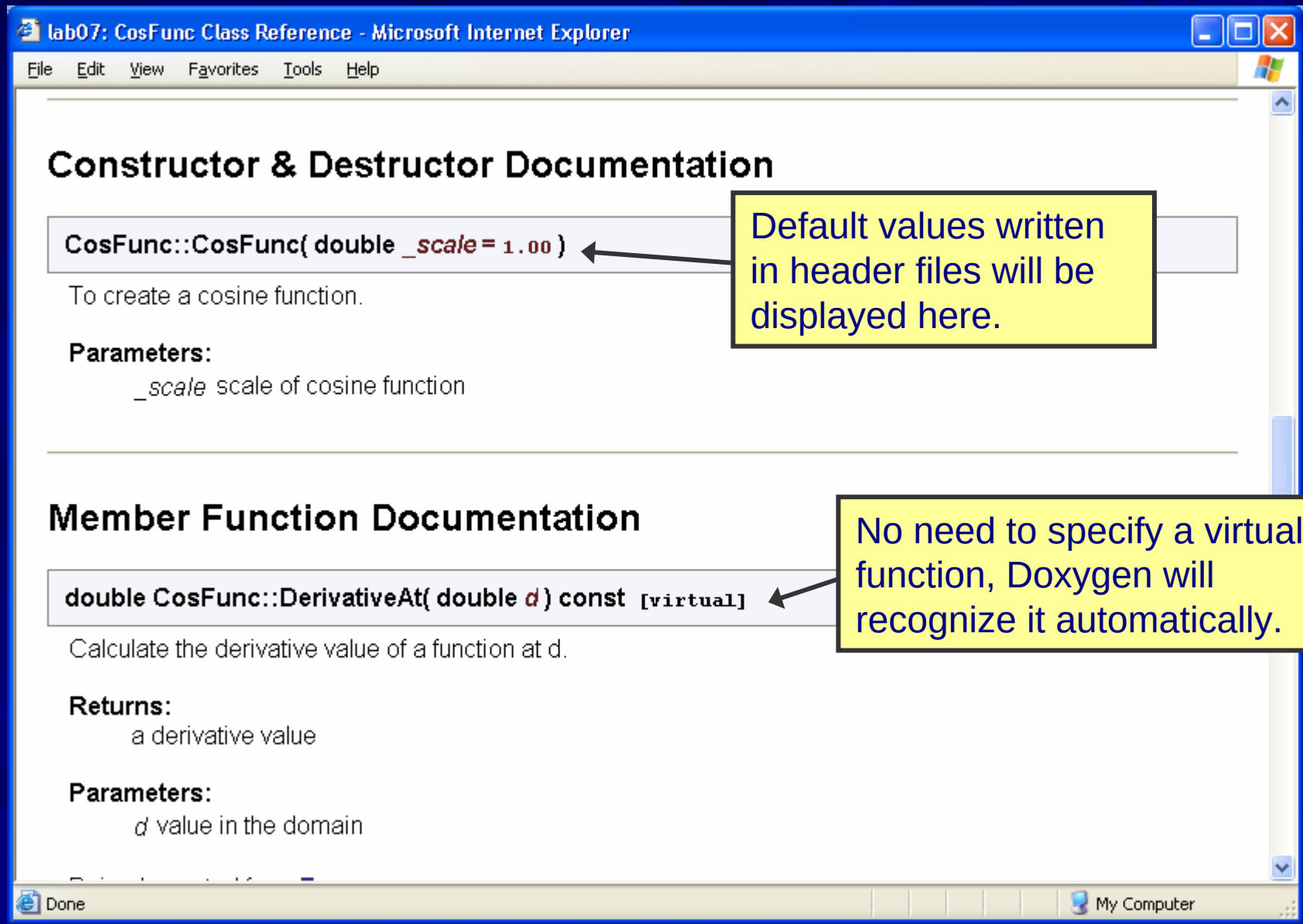
\return

- You can omit \return for methods that return “void” and for constructors.
  - But include it for all other methods, even if its content is entirely redundant. That will make it easier for someone to find the return value quickly.
- Whenever possible, supply return values for special cases, e.g.
  - specifying the value returned when an out-of-bounds argument is supplied.

Parameters

```
double EvaluateAt( double d     //!< value in the domain
                                ) const;
…
double scale;      //!< Scale of the cosine function
```

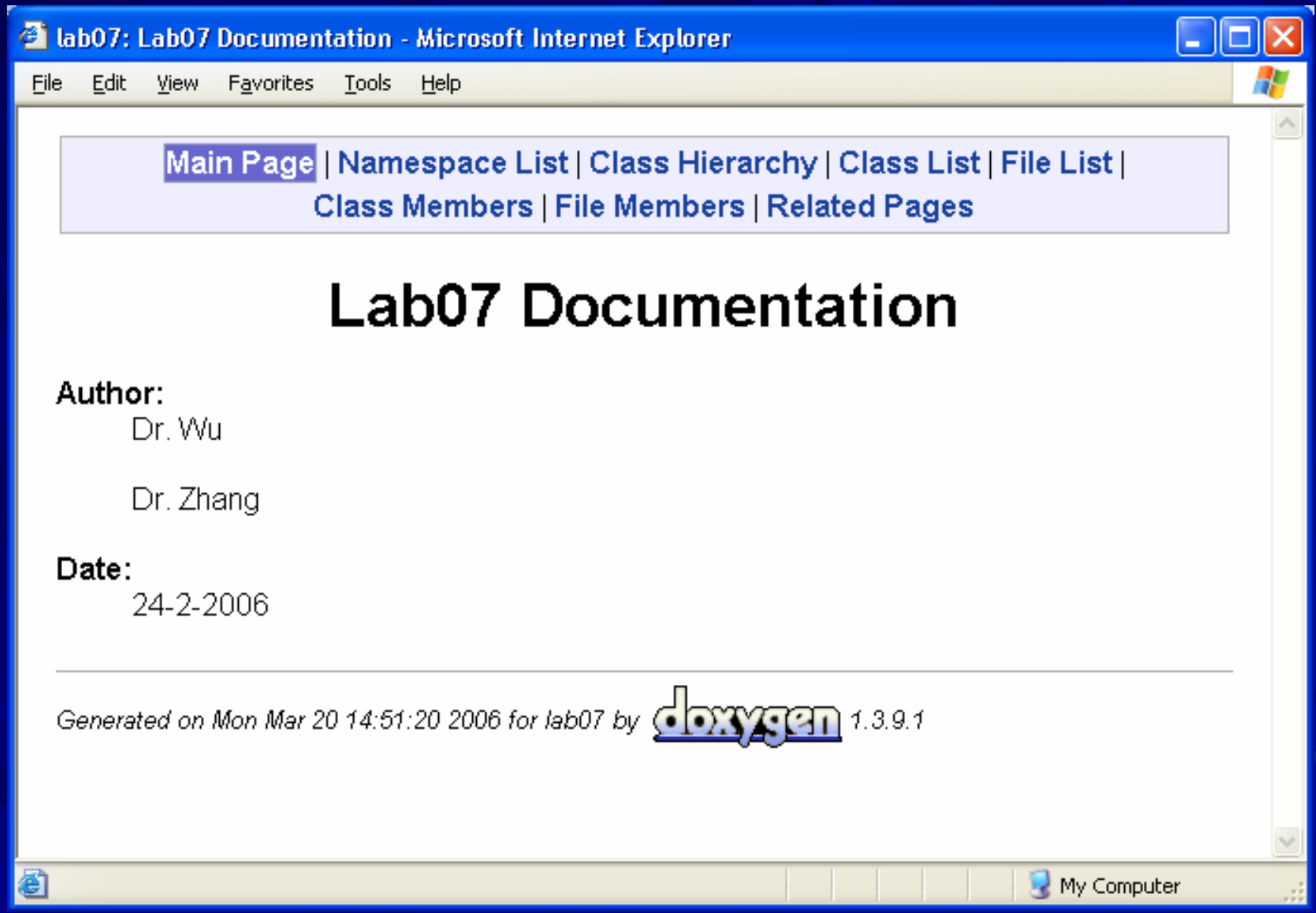
- When documenting the members of a class, or parameters of a function:
  - It is desired to place the documentation block after the member instead of before.
  - For this purpose you have to put an additional < marker in the comment block, i.e. “//!<“


lab07: CosFunc Class Reference - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Constructor & Destructor Documentation
CosFunc::CosFunc(double _scale = 1.00)
To create a cosine function.
Parameters:
_scale scale of cosine function
Default values written in header files will be displayed here.
Member Function Documentation
double CosFunc::DerivativeAt(double d) const [virtual]
Calculate the derivative value of a function at d.
Returns:
a derivative value
Parameters:
d value in the domain
No need to specify a virtual function, Doxygen will recognize it automatically.
Done
My Computer

Documentation for main page

```
/**
 * \mainpage Lab07 Documentation
 *
 * \author  Dr. Wu
 * \author  Dr. Zhang
 * \date    24-2-2006
 */
```

- \mainpage [(title)]
  - It is used to customize the index page.
  - Title argument is optional and replaces the default title that Doxygen normally generates (the one you specified in doxygen.config).
  - You can put it before a class or above the main().
- \author, \date and some related commands can also be used in other documentation blocks.


lab07: Lab07 Documentation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Main Page | Namespace List | Class Hierarchy | Class List | File List |
Class Members | File Members | Related Pages
Lab07 Documentation
Author:
Dr. Wu
Dr. Zhang
Date:
24-2-2006
Generated on Mon Mar 20 14:51:20 2006 for lab07 by doxygen 1.3.9.1
My Computer

Suggested order for a list of commands

- \brief
- \param
- \return
- \exception
- \author
- \version
- \date
- \see
- \since
- \bug
- \warning

Other useful commands

- \note { text }
  - Starts a paragraph where a note can be entered. The paragraph will be indented.
  - A “Note: “ will appear in the documentation before the text.
  - E.g. *note* Impulse function is differentiable except at the impulse

- \par [(paragraph title)] { paragraph }
  - If a paragraph title is given this command starts a paragraph with a user defined heading.
  - The paragraph following the command will be indented.
  - E.g

    par Format:

    name = sin \n

    is periodic = true \n

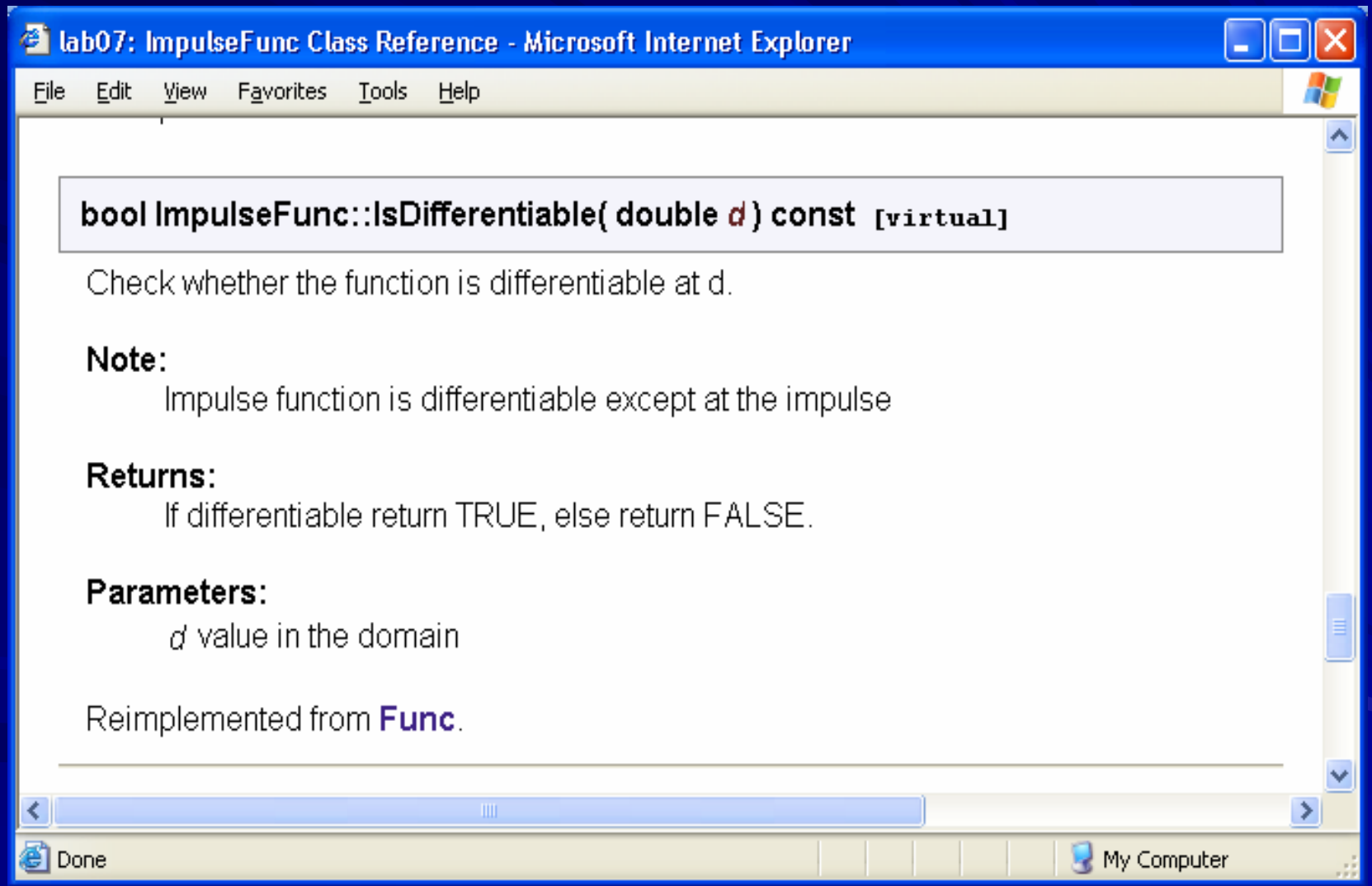
    …

Other useful commands (con't)

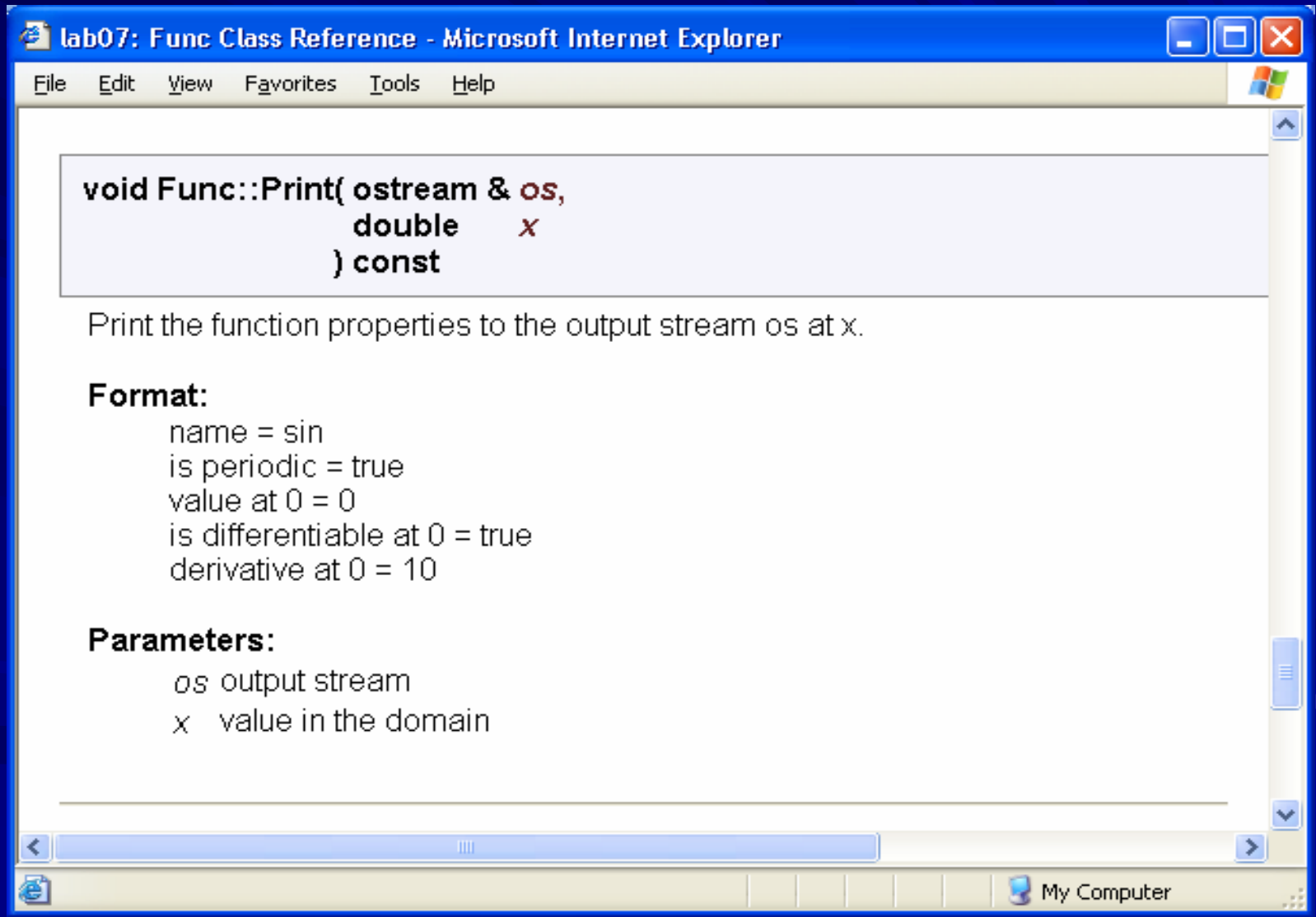
- \n
  - Forces a new line. Equivalent to
 and inspired by the printf function.
  - If no \n, new-lines, tabs and spaces will be trimmed to a single space in the documentation.

- There are other commands for formatting, e.g.
  - \b (bold)
  - \e (italic)
  - \c (type writer)

\note


lab07: ImpulseFunc Class Reference - Microsoft Internet Explorer
File Edit View Favorites Tools Help
bool ImpulseFunc::IsDifferentiable(double d) const [virtual]
Check whether the function is differentiable at d.
Note:
Impulse function is differentiable except at the impulse
Returns:
If differentiable return TRUE, else return FALSE.
Parameters:
d value in the domain
Reimplemented from Func.
Done
My Computer

\par and \n


lab07: Func Class Reference - Microsoft Internet Explorer
File Edit View Favorites Tools Help
void Func::Print(ostream & os,
double x
) const
Print the function properties to the output stream os at x.
Format:
name = sin
is periodic = true
value at 0 = 0
is differentiable at 0 = true
derivative at 0 = 10
Parameters:
os output stream
x value in the domain
My Computer

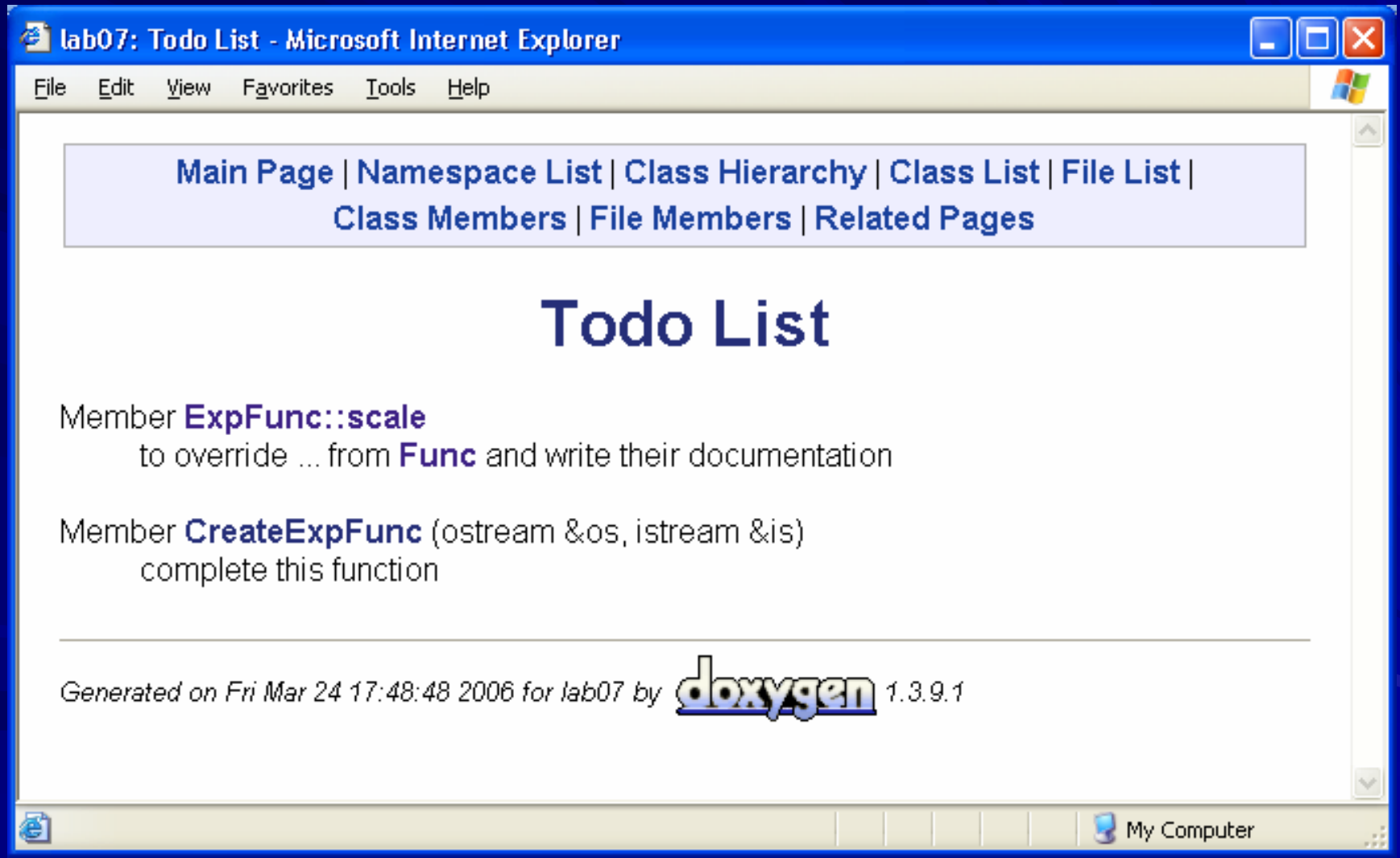
Other useful commands (con't)

- \todo { paragraph describing what is to be done }
  - Starts a paragraph where a TODO item is described.
  - The description will also add an item to a separate TODO list.
  - Those descriptions will be cross-referenced.

- You can find a TODO list in the "Related Pages" of the lab7 documentation.

- You can find other Doxygen commands here:
  - Online reference for special commands:
    - http://www.stack.nl/~dimitri/doxygen/commands.htm

\todo


lab07: Todo List - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Main Page | Namespace List | Class Hierarchy | Class List | File List |
Class Members | File Members | Related Pages
Todo List
Member ExpFunc::scale
to override ... from Func and write their documentation
Member CreateExpFunc (ostream &os, istream &is)
complete this function
Generated on Fri Mar 24 17:48:48 2006 for lab07 by doxygen 1.3.9.1
My Computer