KUTZTOWN UNIVERSITY

Department of COMPUTER SCIENCE

Documentation Standards

Robust, understandable, and well written documentation is a central facet of our discipline. A student who cannot write well is not prepared to fulfill our department's objectives for our graduates.

Documenting Programming Assignments

Files

All files, whether containing code, scripts, or other aspects of a program assignment, e.g. a readme, diagram, manual, etc; must contain an identification comment (ID) block at the very top . At a minimum, the ID block must contain the following items:

- Author – full name of the author
- Major – author's major
- Creation Date – date original file was created
- Due Date – date project is due
- Course – course prefix, number and section for the file
- Professor Name – name of the course instructor
- Assignment number – the assignment number
- Filename – the name of the file
- Purpose – a brief, high-level description of the file

The professor may instruct you on additional items to be included or formatting specifications to follow.

Below is an example of the required information for the identification comment block.

```
/************************************************************/
/*   Author:            Joe Student                         */
/*   Major:             Information Technology              */
/*   Creation Date:     February 18, 2013                   */
/*   Due Date:          March 3, 2013                       */
/*   Course:            CSC135 010                          */
/*   Professor Name:    Dr. Spock                           */
/*   Assignment:        #1                                  */
/*   Filename:          test.cpp                            */
/*   Purpose:           This program will accept the number */
/*                      of PCs, memory, disks and software  */
/*                      and print a bill of sale.           */
/************************************************************/
```

The *'d border is not required, but proper formatting of comment blocks is expected. Develop a style you are comfortable with and that can be read and interpreted easily. It is recommended that you use comments styled for the documentation tool for your language, e.g. Javadoc or Doxygen style comments.

The purpose/description must be complete and robust. Stating that a header file of a class is the class declaration/definition, or a class' cpp file is the implementation file provides nothing of value to the reader.

The ability to write in a manner that is clear, thorough and grammatically-correct is an indispensable skill that you must develop.

Documentation Standards

Function / Method Documentation

Each function or method you write must also be documented. At a minimum, the function / method documentation must include the following items:

- Function or method name
- Purpose/Description of the function or method
- Parameters – list each parameter, specifying its name, its type (input or import, output or export, input/output or import/export) and a brief description
    - If the function / method does not have any parameters, specify *none* or *N/A*
- Return value – specify a brief description of the return value
    - If the function / method does not return a value, specify *none* or *N/A*

Your instructor may inform you of additional items to be included or formatting specifications to follow.

Below are two examples of the required information of function / method documentation.

```
/*************************************************************************/
/*                                                                       */
/*   Function name:      calcTotal                                       */
/*   Description:        Calculate total owed for a stay in the hospital.*/
/*   Parameters:         int   num_days: days in hospital - input        */
/*                       char  room_type: type of room - input           */
/*                       bool phone_use: was telephone used? – input     */
/*   Return Value: double - the total amount owed                        */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*   Function name:      getAge                                          */
/*   Description:        Provides the age of the person.                 */
/*   Parameters:         none                                            */
/*   Return Value:       int - the person’s age                          */
/*                                                                       */
/*************************************************************************/
```