C Coding Conventions

1.Introduction

This document is intended to outline the C coding conventions for the homeworks of the class CSE102. These conventions will provide a standard and consistent look and feel to the source code generated by students. This document will also serve as a template and reference for source code files. This should ensure that required features like necessary comment blocks are included in all files.

The conventions in this document must be followed by all students and will be enforced with a penalty of at least 10% of the total grade for a homework.

For all examples in this document, a student with the number 091041004 and name "Ali Ahmet Veli" is considered. Replace with your student number and name when working with your files. You should not use non-English characters in any part of your homework (including file names, comments, etc.).

2.File Naming

•For homeworks in CSE102, file names will be restricted to the following format:
HW##_<StudentName(and 2nd name if exists)>_<StudentSirname>_<StudentNumber>.c (same for .h files) (replace ## and # with the proper numbers). Use exactly **3** '_' symbols.

For example, for the homework 5 of CSE102, the source code file will be named like this:
HW05_AliAhmet_Veli_091041004.c

•Unless otherwise asked, submit your source file(s) **only** (no data files, executables, etc.) and do NOT COMPRESS the file(s) you submit.

3.Source File Formats (Both .c and .h)

•A line of text in a source file must not exceed **80 characters**. This requirement is intended to prevent the text from wrapping.

In gedit, go to *Edit->Preferences*. In the *View* tab, select *Display right margin* and choose column 80. This will help you by showing the 80-th column with a vertical line, so that you may break a line before the 80-th column.

•Indenting with in the source file will be **4 spaces**.

In gedit, the tab key on the keyboard can be directly used to put 4 spaces. To enable this option, go to *Edit->Preferences*. In the *Editor* tab, enter *Tab width* as 4 .

•In most of homeworks in this course you will write only 1 source file having ".c" extension, but in some of your homeworks you will also write header files for function prototypes.

•A source file (.c file) will follow the outline listed below.

- File comment box
  - Name of the file
  - Creation date
  - Author's name and number
  - Description of the file
  - Notes (if any)
  - Referenced technical information
- Include section
- Constant definition section (#defines goes here) (Optional)
- Type definition section (typedefs goes here) (Optional)
- Macro section (Optional)
- Function prototype section (a comment box for each function). If you also provide a header file, this section should be prepared for each static function. Otherwise, this section should be prepared for each function.
  - Function name
  - Definition of parameters
  - Definition of the return value
  - Description of the function
  - Notes (if any)
- Function section
  - Brief description of functions
  - Function bodies (code of functions)
- End of file comment

•A source header file (.h file) will follow the outline listed below.

- File comment box
  - Name of the file
  - Creation date
  - Author's name
  - Description of the file
  - Notes (if any)
  - Referenced technical information
- #ifndef section to prevent multiple includes
- Include section
- Constant definition section (#defines goes here) (Optional)
- Type definition section (typedefs goes here) (Optional)
- Macro section (Optional)
- Global variable declaration section (Optional and should generally be omitted)
- Function prototype section (a comment box for each function)
  - Function name
  - Definition of parameters
  - Definition of the return value

    - Description of the function
    - Notes (if any)
- #endif (close #ifndef)
- End of file comment

4.Coding

•***Constants***

All upper case letters. Use _ to separate words.

```
#define MAX_RATIO 0.5
const int PI = 3.14159265359;
```

•***Parameterized macros***

All upper case letters. Use _ to separate words. Use parenthesis to protect arguments.

```
#define MIN(a, b)     ( (a) < (b) ? (a) : (b) )
```

•***Type/Variable/Function Names***

Start each word with lower-case letters in the type name. Use underscores to separate words.

```
max_so_far
repeat_count
pixel
byte8
linked_list_node
```

5.Indentation

- The contents of function, repetition (for, while, do..while), selection (if, switch), definition blocks should be indented. You may open parenthesis in the current line or in a new line but close in a separate line. You may omit parenthesis in 1-line selection or repetition blocks.

```
if ( <condition> )
{
      <code>
}
else if ( <condition> )
{
      <code>
}
else
{
      <code>
}

or

if ( <condition> ){
      <code>
}
else if ( <condition> ){
      <code>
}
else{
      <code>
}
```

```
if(x<y)
      do_something();
```

```
while( <condition> )
{
      <code>
}

while( <condition> ){
      <code>
}
```

- If the closing } bracket is separeted from the opening { bracket by a distance not easily displayed, then the closing } bracket should identify the initial statement.

```
while( <condition> )
{
      ...
      <a lot of lines of code>
      ...
} /* End of while( <condition> ) */


switch(<expression>
{
      case <const>:
            {
                  ...
                  <a lot of lines of code>
                  ...
            } /* End of case <const> */
            break;

      case <const>:
            <code>
            /* Fall through */

      case <const>:
            <code>
            break;

      ...

      default:
            <code>
}
```

6.Functions

- To increase the observability of the function names, write the return-type of the functions in a separate line as below.

```
<return-type>
<function-name> ( <parameters> )
{
      <declerations>
      <body>
}
```

<declerations>

- Unless they naturally form a group, define each variable in a separate line. Comment each variable/group on its line (loop variables may be left uncommented).

```
double average;   /* Average grade of the students */
int student_count;      /* Total number of students */
double angle_a, angle_b, angle_c; /*angles of the triangle*/
int i, j;
```

- You should always use meaningful variable names (they should remind the meaning of the value they hold).
- Parameters: Variable naming convention applies to parameters.

7.Comments

- You should write comments on the following cases:
  - at file comment boxes
  - at function prototype section comment boxes (full description of the function)
  - at function section comment boxes (brief description of the function)
  - at variable definitions
  - inside your code: explain your code briefly as below

```
/*explanation for code block 1*/
<code block 1>

/*explanation for code block 2*/
      <code block 2>
            ….      /*a short explanation for this line*/
      /*a large explanation for the next line …..      */
      <code line to be explained>
```

- Avoid unnecessary/unnecessarily large/nonsense comments.

8. C Standart

- In this course, we accept **ANSI C89** standard in which following programming features are forbidden:
  - Variable-length arrays: Array capacities should be given as constants. Only dynamic memory allocation on heap is supported (using malloc, etc.)
  - Mixed declaration: All variable declarations should be done before any other lines of code.
  - One line comments beginning with “//”
- Our test environment is **Linux** operating system using **gcc** compiler with following options: **-ansi, -pedantic-errors**

9.Example

a.Header file

```
/*############################################################################*/
/*                                                                            */
/* HW05_Name_Sirname_091041004.h                                              */
/* ---------                                                                  */
/* Created on 28/02/2010 by Alparslan YILDIZ                                  */
/* Modified on 09/02/2015 by Evren Cifci                                      */
/*                                                                            */
/* Description                                                                */
/* -----------                                                                */
/*    The HW05_Name_Sirname_091041004.h contains functions for                */
/*    the homework II, which are used for drawing a shape.                    */
/*                                                                            */
/* Notes                                                                      */
/* -----                                                                      */
/* This is a demo files. Usually, the text with in the prolog is more         */
/* verbose.                                                                   */
/*                                                                            */
/* References                                                                 */
/* ----------                                                                 */
/* (If any)                                                                   */
/*                                                                            */
/*############################################################################*/

#ifndef _HW05_091041004_PART1_H_
#define _HW05_091041004_PART1_H_

/*----------------------------------------------------------------------------*/
/*                                Includes                                    */
/*----------------------------------------------------------------------------*/
#include <stdio.h>

/*----------------------------------------------------------------------------*/
/*                                 #defines                                   */
/*----------------------------------------------------------------------------*/
#define    PI                3.14
#define    MAX_EDGE_LEN        16

/*----------------------------------------------------------------------------*/
/*                                  typedefs                                  */
/*----------------------------------------------------------------------------*/
typedef unsigned char        byte;
typedef struct
{
    byte        pix;
    int         depth;
}pixel;

/*----------------------------------------------------------------------------*/
/*                         Function Prototypes                                */
/*----------------------------------------------------------------------------*/

/*############################################################################*/
/*                                                                            */
/* int draw_square(int ox, int oy, int a)                                     */
/* -----------                                                                */
/*                                                                            */
/*     ox - x coordinate of the center of the square                          */
/*     oy - y coordinate of the center of the square                          */
/*      a - Edge length of the square                                         */
/*                                                                            */
/* Return                                                                     */
/* ------                                                                     */
/*     0 on success                                                           */
/*     -1 on error                                                            */
/*                                                                            */
/* Description                                                                */
/* -----------                                                                */
/*     This function takes the center and edge length of a square and draws  */
/*     it on the screen if the center coordinates are inside the screen and  */
/*     edge length does not exceeds screen limits.                            */
/*                                                                            */
/* Notes                                                                      */
/* -----                                                                      */
/*     The user is responsible that the screen is clear at the time of call. */
/*                                                                            */
/*############################################################################*/
int
draw_square(int ox, int oy, int a) ;

#endif

/*############################################################################*/
/*                       End of HW05_Name_Sirname_091041004                   */
/*############################################################################*/
```

b. Source file

```
/*###########################################################################*/
/*                                                                           */
/* HW05_Name_Sirname_091041004_part1.c                                       */
/* ---------                                                                 */
/* Created on 28/02/2010 by Alparslan YILDIZ                                 */
/* Modified on 09/02/2015 by Evren Cifci                                    */
/*                                                                           */
/* Description                                                               */
/* -----------                                                          */
/*    The HW05_Name_Sirname_091041004.c contains the implementations of */
/*    functions for the first part of the homework II, which are used for */
/*    drawing a shape.                                                  */
/*                                                                      */
/* Notes                                                                */
/* -----                                                                */
/* This is a demo files. Usually, the text with in the prolog is more   */
/* verbose.                                                             */
/*                                                                      */
/* References                                                           */
/* ----------                                                           */
/* (If any)                                                             */
/*                                                                      */
/*######################################################################*/

/*----------------------------------------------------------------------*/
/*                               Includes                               */
/*----------------------------------------------------------------------*/
#include <stdlib.h>
#include "HW05_Name_Sirname_091041004.c"

/*----------------------------------------------------------------------*/
/*                         Function Prototypes                          */
/*----------------------------------------------------------------------*/

/*#######################################################################*/
/*                                                                       */
/* int draw_pixel(int x, int y, char pen)                                */
/* -----------                                                           */
/*                                                                       */
/*      x   - x coordinate of the pixel                                  */
/*      y   - y coordinate of the pixel                                  */
/*      pen - the character pen will be used to draw the pixel           */
/*                                                                       */
/* Return                                                                */
/* ------                                                                */
/*     0 on success                                                      */
/*     -1 on error                                                       */
/*                                                                       */
/* Description                                                           */
/* -----------                                                           */
/*     This function takes the position of a pixel and draws it on the   */
/*     screen if the pixel coordinates are inside the screen and pen is a */
/*     valid character.                                                  */
/*                                                                       */
/*#######################################################################*/
static int
draw_pixel(int x, int y, char pen) ;


/*----------------------------------------------------------------------*/
/*                       Function Implementations                       */
/*----------------------------------------------------------------------*/

/* Function draw_pixel                                                   */
/* -----------                                                              */
/*     This function takes the position of a pixel and draws it on the      */
/*     screen if the pixel coordinates are inside the screen and pen is a    */
/*     valid character.                                                     */
static
int draw_pixel(int x, int y, char pen)
{
    ...
    <code of the function>
    ...
}

/* Function draw_square                                                     */
/* -----------                                                              */
/*     This function takes the center and edge length of a square and draws  */
/*     it on the screen if the center coordinates are inside the screen and  */
/*     edge length does not exceeds screen limits.                          */
int
draw_square(int ox, int oy, int a)
{
    ...
    <code of the function>
    ...
}


/*##########################################################################*/
/*                       End of HW05_Name_Sirname_091041004.c               */
/*##########################################################################*/
```