

Style Guidelines

Compile code using (**without error or warning**):

```
gcc -std=c99 -Wall -pedantic -o
```

Coding conventions for this class:

Standards

- › Braces on separate lines.
- › Indents (4 spaces) Everything between { and } should be indented.
- › Do NOT add extra indenting to statements that are not inside another block for the purposes of making the code "pretty".
- › Main must return an int.

Comments

- › Comments for block go inside block.
- › `/*style comments*/` are required for beginning of functions and program
- › C++ style comments can be used at ends of lines
- › Switch statements - braces next line - blank line after a break.

Variables

- › Variable and function names are in camelCase.
- › Single letter variables allowed only for loop counters.
- › Variables are declared at beginning of a function.
- › Variables are initialized separately from being declared.

```
int i;  
i=0;
```

- › All functions must be prototyped at the top of the file and defined at the bottom of the file ie:

```
/*  
*****  
CIS 1500 Style Example  
Justin Carvalho  
Nov 1/13  
This program is a demonstration of correct  
programming style for cis1500  
***** */
```

Style Guidelines

```

/*****
CIS 1500 Style Example
Justin Carvalho
Nov 1/13
This program is accepted style for cis1500
*****/

#include <stdio.h> //Used for printf
#include <stdlib.h>

void printMsg (int msg);
int squareFunction (int x);

int main()
{
    int variableX; //Variables should be declared
    variableX = 1; //and initialized on separate lines
    /*Assignments and boolean logic separated by spaces*/
    if (variableX == 1 || variableX == 2)
    { /*new braces on next line*/
        printf("This is inside 2 blocks\n");
        printf("So it gets 2 indents\n");
    }
    else
    {
        printf("Me too\n");
    }
    printMsg(variableX);
    squareFunction(variableX);
    return 0;
}

/*****
PrintMsg: Prints a message based on $msg
In: int msg
Out: None
Post: Message gets printed
*****/
void printMsg (int msg)
{
    int i;
    printf("Hello world\n");
    for (i = 0; i < 2; i ++)
    {
        switch(msg)
        {
            case 1:
                printf("Case lines up with switch\n");
                break; //Blank line after break

            case 2:
                printf("No blank line if no break\n");
            case 3:
                printf("Multi-line print statements\n" \
                    "On different lines. But should line\n" \
                    "up with the rest of the string\n");
                break;

            default:
                printf("All cases have a default\n");
        }
    }
    return;
    /*Not neccessary, but void functions should still
    say a return.*/
}

/*****
Square: Squares an int
In: int x
Out: int
Post: x is squared and returned
*****/
int squareFunction(int x)
{
    x = x * x;
    /*Math functions seperated by spaces*/
    return x;
}

/*Blank new line at end of file*/
```