

**Engr B19c/c++ Shop Standards:**

**(1) General Guidelines**

✓ Avoid excessive blank lines and spaces. ✓ Only include relevant and informative comments. ✓ Do not wrap sentences on the execution screen that the user views. ✓ Make sure words printed to the monitor are spelled correctly. ✓ Do not wrap programming statements on the printed hard copies. ✓ Do not use mixed mode expressions that cause data promotion.

**(2) Preprocessor Directives**

✓ The #include statements should be above main and below header documentation. ✓ Do not use .h header files and include using namespace std; ✓ # define statements should be listed next. ✓ Function prototypes should be listed next separated by one blank line.

**(3) Function main**

✓ For function main, include the header documentation with description, input & output.

/\*

your name Engr B19c/c++ program number date

Program description –

Input: Output: \*/

✓ Function main should be the first function in your file. ✓ Must have the system (“pause”); (for Dev C++ compiler) and return 0; statements at the end

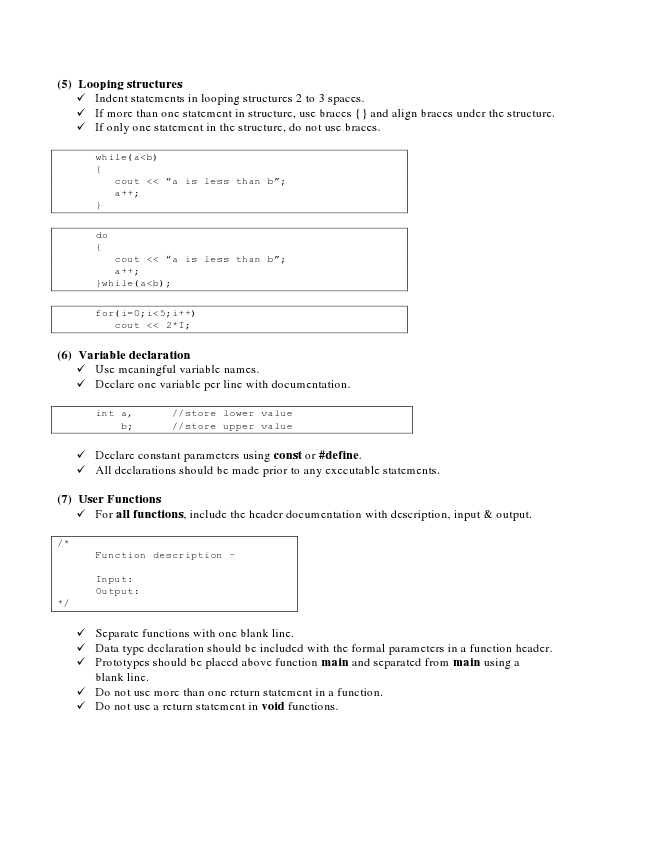
of main.

**(4) Selection structures**

✓ Indent statements in decision structure 2 to 3 spaces. ✓ If more than one statement in the structure, use braces {} and align braces under the structure. ✓ If only one statement in the structure, do not use braces.

if (a < b) b=a; else {

a=b; b=a\*200; }



**(5) Looping structures**

✓ Indent statements in looping structures 2 to 3 spaces. ✓ If more than one statement in structure, use braces {} and align braces under the structure. ✓ If only one statement in the structure, do not use braces.

while(a<b) {

cout << “a is less than b”; a++; }

do {

cout << “a is less than b”; a++; }while(a<b);

for(i=0;i<5;i++)

cout << 2\*I;

**(6) Variable declaration**

✓ Use meaningful variable names. ✓ Declare one variable per line with documentation.

int a, //store lower value b; //store upper value

✓ Declare constant parameters using const or #define. ✓ All declarations should be made prior to any executable statements.

**(7) User Functions**

✓ For all functions, include the header documentation with description, input & output.

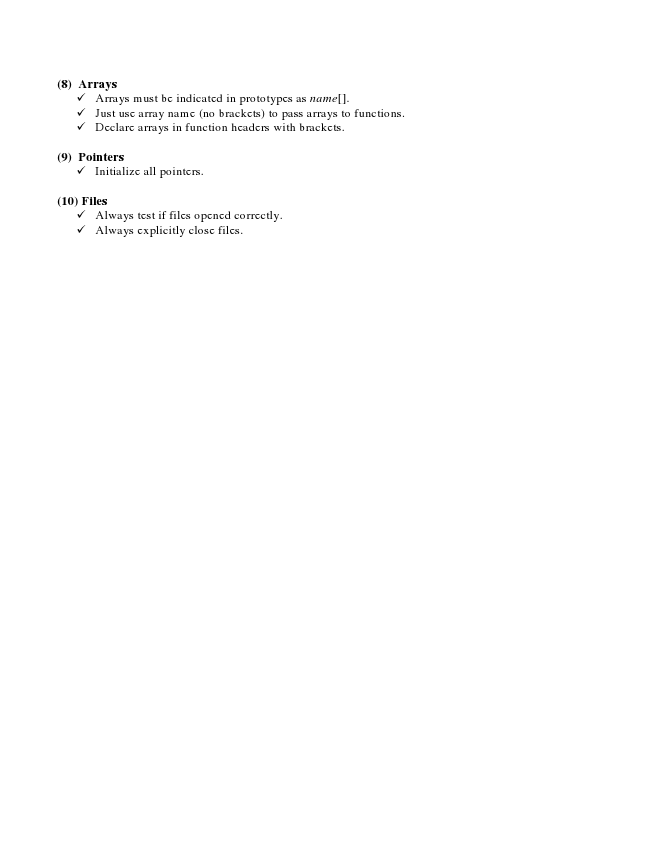
/\*

Function description –

Input: Output: \*/

✓ Separate functions with one blank line. ✓ Data type declaration should be included with the formal parameters in a function header. ✓ Prototypes should be placed above function main and separated from main using a

blank line. ✓ Do not use more than one return statement in a function. ✓ Do not use a return statement in void functions.



**(8) Arrays**

✓ Arrays must be indicated in prototypes as name[]. ✓ Just use array name (no brackets) to pass arrays to functions. ✓ Declare arrays in function headers with brackets.

**(9) Pointers**

✓ Initialize all pointers.

**(10) Files**

✓ Always test if files opened correctly. ✓ Always explicitly close files.