**MY CODING STANDARDS**

Listed below are the coding practices that I will follow:

1. **Variable Naming**

snake case letter will be used to denote the variable name,and first letter always will be start with capital letter and in my variable naming I use short form of data type to easily identify variable data type at any place I use “In” for int, “Ch” for char “Fl” for float and so on. Data type short name use in starting of variable with underscore to define a variable name.

**Eg:**  int In\_age; char Ch\_letter; float Fl\_value; double D\_value;

1. **Function Naming:**

camel case will be used to define functions, I.e the 1st letter will be small and the 1st letter of the next words will be capital.

**Eg:** void addFun(); int addFun();

1. **Indentations:**

space of one tab will be left when there is the use of ‘{‘, plus any other time there is a need to separate the coming part from the general flow of control like calling a function. And use the inline comment in end of the control statement that can easy to understand where any control statement is ending.

**Eg:** for(int i=0; i<10; i++){

int In\_a=0; // initialize a

a=i; //assign the value

} //end of for loop

1. **Comments:**

In my coding standard I use single line comment use to define the end of loop and end of the statement braces that help to understand that where is a statement ending and I will use multi-line comments to define a control statement , function and methods because it is very helpful to understand the work of any control statement, function and method.

**Eg:**

// create a function to calculate sum of a and b

int myFunction(int a, int b) {

int In\_a1 = a;

int In\_b2 = b;

return a1+b2;

}

int main() {

int sum = myFunction(10,20); // call the function

printf("%d",sum);

return 0;

}

1. **Sequencing:**

the sequencing of the program will be as follows:

1. The copyright statement
2. #include commands
3. Function if need.
4. main function-
5. variable declaration and use the statements like control statement, conditional statement.

This sequence will ensure a uniform approach to the program that can be easily understood in later stages.