Rules of "Coding Standards"

* Use one single "TAB" (and not 3-4 spaces) for indentation.

*Indentation means, whatever is logically 'inside' should be shown like that. This means the 'bodies' of for, while, if, else, function, switch should be indented using ONE tab.

* Indent when you are typing the code and not after writing the code.

*** Rules for spacing:**

Give one space after comma in variable lists or arguments lists.

e.g int i, j, k;

OR printf("%d%d", a, b);

Give one space before and after every binary operator: e.g.   i = j + k;

*Everything which has a { and } associated with it is called a block.

The opening and closing brackets of any "block" must be matched with the first character of the block keyword (i.e. f of for, i o f if, e of else, etc.)

* Do not give a space before ";" of a for loop

* Do not give a space after '(' and ')'

/* */ is a C comment, // is a C++ comment. Use /* */ only.

All variable declarations should be in the beginning of the code, do not declare variables anywhere in the code (Unless you are sure you want to do it).

Wrong:
```
int i , j , k;
```
Correct:
```
int i, j, k;
```
Comment:
Space only after the comma in argument, variable list, not before it.

Wrong:
```
for(i = 0 ; i < 10 ; i++)
```
Correct:
```
for(i = 0; i < 10; i++)
```
Comment:
Space only after the ';' in argument, variable list, not before it.

Wrong:
```
for ( i = 0; i < 10; i++)
```
Correct:
```
for(i = 0; i < 10; i++)
```
Comment:
No space after for, and after (or before)

Wrong:
```
int main() {
    int i, j, k;
        for(i = 0; i < 10; i++)
            printf("hi\n");
}
```
correct:
```
int main() {
    int i, j, k;
    for(i = 0; i < 10; i++)
        printf("hi\n");
}
```
Comment:
Both "for" and "int i,..." are "inside main", so only 1 level of indentation.

Wrong:
```
int main()
    {
        int i, j, k;
        printf("hi\n");
    }
```
Correct:
```
int main() {
    int i, j, k;
    printf("hi\n");
}
```
Comment:
Bracket } must match the "i" of int. The wrong code has double indentation.

Wrong:
```
scanf("%d%d",&i,&j);
```
Correct:
```
scanf("%d%d", &i, &j);
```
Comment:
one space needed after every argument in an argument/variable list.

Wrong:
```
scanf("%d %d", &i, &j);
```

Correct:
```
scanf("%d%d", &i, &j);
```
Comment:
```
There should be no space in %d%d of scanf.
```

Wrong:
```
printf("\n hi there");
```
Correct:
```
printf("hi there\n");
```
Comment:
Newline after printf. Reasons a) Prompt comes back properly b) It flushes the output faster to screen.

Wrong:
```
int main() {
    int i, j, k;
return 0;
}
```
Correct:
```
int main() {
    int i, j, k;
    return 0;
}
```
Comment:
return is "inside" main(), so use one indentation level.

Wrong:
```
void main() {
}
```
Correct:
```
int main() {
}
```
Comment:
main() should return int always.

Wrong:
```
if(i > 0) {
printf("hi\n");
}
else {
printf("bye\n");
}
```
correct:
```
if(i > 0) {
    printf("hi\n");
}
else {
    printf("bye\n");
```

```
}
```
Comment:
Always indent what is "inside" if or else.

Wrong:
```
switch(x) {

case 'a':

printf("hi\n");

break;

case 'b':

scanf("%d", &i);

break;

}
```
correct:
```
switch(x) {

     case 'a':

          printf("hi\n");

          break;

     case 'b':

          scanf("%d", &i);

          break;

}
```
Comment:
Indent both 'case' and the code of case statement in a switch statement.

Some other important things (not related to formatting, indentation)

* Main is always "int main"

* Write a \n at the end of printf

* Do not have any \n or space in scanf

* Do not use Windows

* Begin a "block" with { on the same line as the block-word (for, if, while, etc)

* Remove all warnings from your code.