

Coding Style

Embedded Software

Wolfgang Neff

Coding Style

- Why using a coding style?
 - Improves the readability
 - Helps to detect errors
 - Gives a uniform appearance
 - Allows better collaboration
 - Just dilettantes don't use it

Naming Conventions (1)

- Use meaningful names

Best Practice

```
int add(int a, int b)
{
    return a+b;
}
//...
int n = add(1,2);
```

Worst Practice

```
int lol(int xxyxx, int xyxx)
{
    return xxyxx + xyxx;
}
//...
int xxxyx = lol(1,2);
```

Naming Conventions (2)

- Start names with small letters

Best Practice

```
int add(int a, int b)
{
    return a + b;
}
//...
int n = add(1,2);
```

Worst Practice

```
int ADD(int A, int b)
{
    return A + b;
}
//...
int n = ADD(1,2);
```

Naming Conventions (3)

- Use camel case for names

Best Practice

```
int myAdd(int pOne, int pTwo)
{
    return pOne + pTwo;
}
//...
int theResult = myAdd(1,2);
```

Worst Practice

```
int MYADD(int Pone, int Ptwo)
{
    return Pone + Ptwo;
}
//...
int theresult = MYADD(1,2);
```

Indentation (1)

- Indent the content of braces

Best Practice

```
int fac(int n)
{
    int p;
    for (int i=0; i<n; i++)
    {
        p = p*i;
    }
    return p;
}
```

Worst Practice

```
int fac(int n)
{
int p;
for (int i=0; i<n; i++)
{
p = p*i;
}
return p;
}
```

Indentation (2)

- Braces start and end where they belong to

Best Practice

```
int fac(int n)
{
|   int p;
|   for (int i=0; i<n; i++)
|   {
|   |   p = p*i;
|   }
|   return p;
}
```

Belongs to for

Belongs to int fac

Worst Practice

```
int fac(int n)
    {
    int p;
    for (int i=0; i<n; i++) {
        p = p*i;
    }
    return p;
}
```

Indentation (3)

- Indentation must be consistent

Best Practice

```
int fac(int n)
{
    ....int p;
    ....for (int i=0; i<n; i++)
    ....{
        .....p = p*i;
    ....}
    ....return p;
}
```

Worst Practice

```
int fac(int n)
{
    ..int p;
    ....for (int i=0; i<n; i++)
    ....{
        ..... p = p*i;
    ....}
    ..return p;
}
```


Indentation (4)

- There are never two braces in one column

Best Practice


```
void func(int n)
{
    for (int i=0; i<n; i++)
    {
        // Do something
    }
}
```

Belongs to for

Belongs to int fac

Worst Practice

```
void func(int n)
{
    for (int i=0; i<n; i++)
    {
        // Do something
    }
}
```



Functions (1)

- Length should not be very large

Best Practice

```
int sqr(int n)
{
    return n * n;
}
double hyp(int a, int b)
{
    int a2 = sqr(a);
    int b2 = sqr(b);
    return sqrt(a2+b2);
}
```

Worst Practice

```
double hyp(int a, int b)
{
    int mya = a;
    int myb = b;
    int a2 = mya * mya;
    int b2 = myb * myb;
    int c2 = a2 + b2;
    double res = sqrt(c2);
    double c = res;
    return c;
}
```

Functions (2)

- They should do one specific task

Best Practice

```
int sqr(int n)
{
    return n * n;
}
double hyp(int a, int b)
{
    int a2 = sqr(a);
    int b2 = sqr(b);
    return sqrt(a2+b2);
}
double x = 2 * hyp(1,2);
```

Worst Practice

```
double hyp2(int a, int b)
{
    int mya = a;
    int myb = b;
    int a2 = mya * mya;
    int b2 = myb * myb;
    int c2 = a2 + b2;
    double c = sqrt(c2);
    double res = 2 * c;
    return res;
}
```