

# 1 Code conventions

```
func(x, y) {  
    ...  
}
```

The indent size is one tab-stop for each level of indenting.

---

```
if (number == 2) {  
    ...  
}
```

```
if ( <komplexExpression> ) {  
    ...  
}  
else {  
    ...  
}
```

```
while (n < 30) {  
    fooBar(n++);  
}
```

Note the convention on function naming and the presence of curly brackets even though there is only one line of code. else always start on it's own line.

---

```
switch (currentToken)  
{  
    case 1:  
    {  
        ...  
        break;  
    }  
    ...  
}
```

---

```
class Foo
{
public:
    void aFunction();

protected:
    int someData;
    void anotherFunction();

private:
    char someMoreData;
    void yetAnotherFunction(data);
};
```

```
Foo *bar = new Foo();
```

- All variables are private or protected; private is preferred.
- All functions should be private or protected if possible.

---

```
//animation.cpp
#include "animation.h"

#include "foofile.h"
...

#include <barfile.h>
...
```

---

Other notes:

- Variables with a long lifetime and large scope such as private variables have long descriptive names such as motionPictureHashMap
- Variables with a shorter lifetime and smaller scope such as lokal or parameter variables have shorter names such as pictureMap
- Loop and short lived test variables have names consisting of one character such as i, j or n.

- Constants (const) are named in capital letters with an underscore between each word: `IMPORTANT_CONSTANT`.