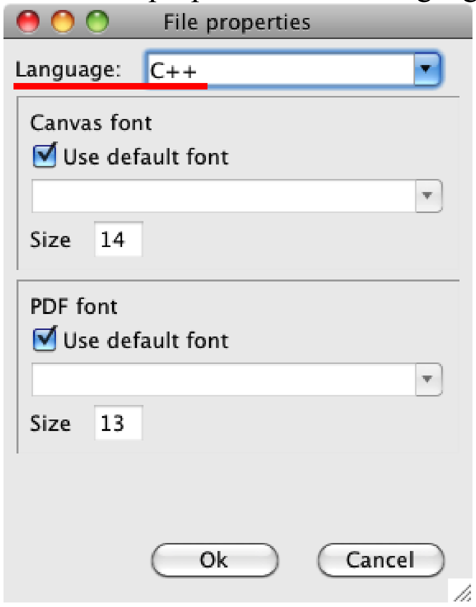
Plain C and C++ programming with DRAKON Editor

1. Set the language to C or C++. File / File properties... → Language.



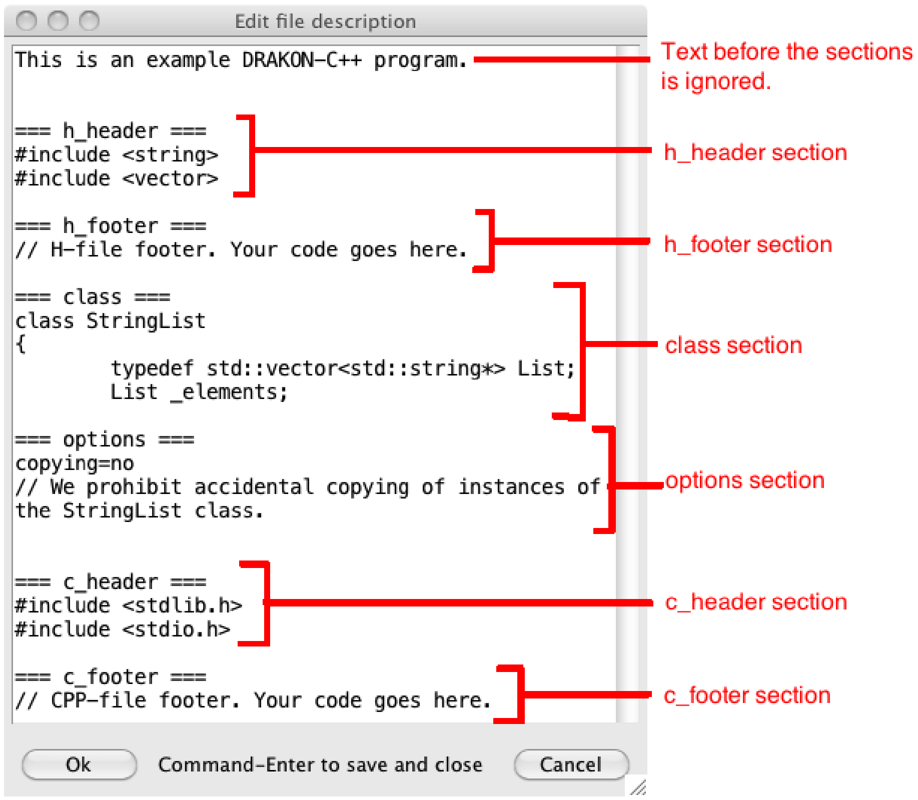
File properties

Language: C++

Canvas font
☑ Use default font
Size 14

PDF font
☑ Use default font
Size 13

2. Add sections to the file description. File / File description...

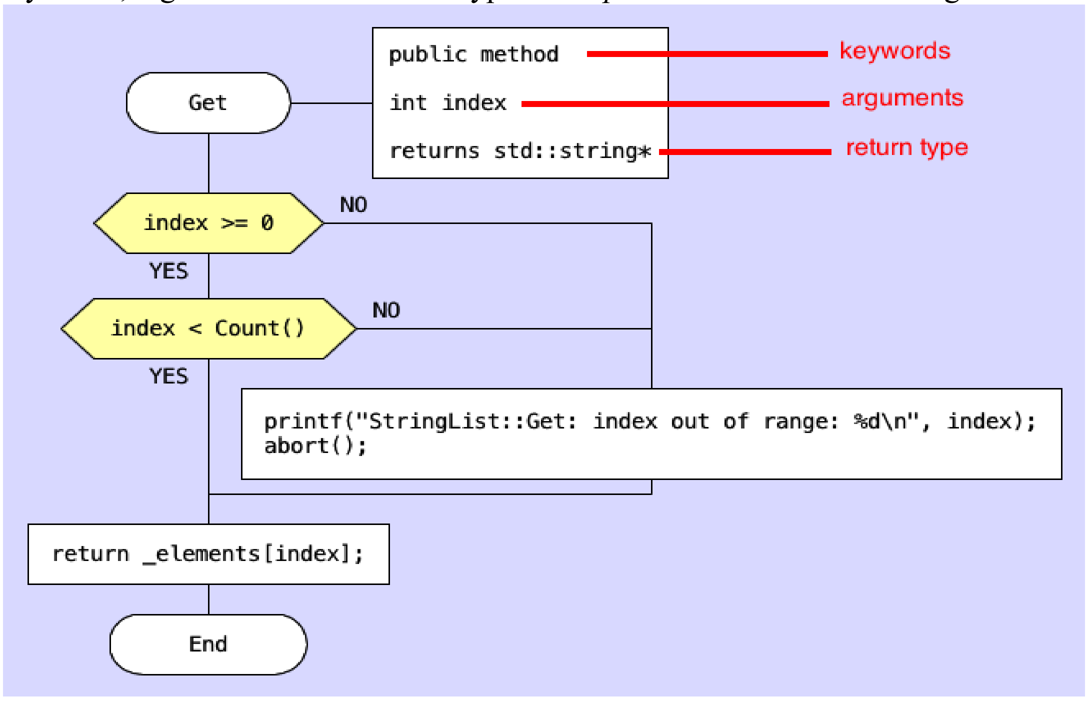


Edit file description

```
This is an example DRAKON-C++ program.


=== h_header ===
#include <string>
#include <vector>

=== h_footer ===
// H-file footer. Your code goes here.

=== class ===
class StringList
{
        typedef std::vector<std::string*> List;
        List _elements;

=== options ===
copying=no
// We prohibit accidental copying of instances of
the StringList class.


=== c_header ===
#include <stdlib.h>
#include <stdio.h>

=== c_footer ===
// CPP-file footer. Your code goes here.
```

All sections are optional.
Sections can be placed in arbitrary order.
The text before all sections is ignored.

3. Add keywords, arguments and the return type to the *parameters* item of the diagram.



Keywords are optional.

Keywords for C++

Access: **public, protected, private**.
Dispatch:

virtual – virtual method
static – for methods: a static method; for free functions: a static (internal) function.

Procedure type:

function – a free function
method – a class method
ctr – a constructor.
dtr – the destructor.

Others:

abstract – pure virtual method
const – const method
inline – inline function or methods

Keywords for plain C

Access:

public – the function is visible outside of the .c file
static – the function is internal to the c.file

Others: **inline**

Sections

=== **h_header** ===
Goes to the top of the .h file.

=== **h_footer** ===
Goes to the bottom of the .h file.

=== c_header ===
Goes to the top of the .c or .cpp file.

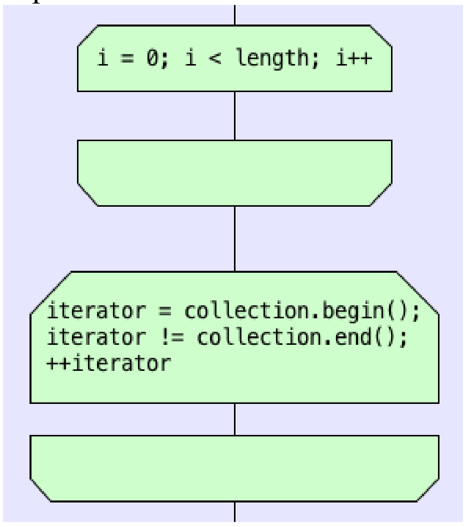
=== c_footer ===
Goes to the bottom of the .c or .cpp file.

=== class ===
C++ only. Contains the beginning of the class declaration.
Must contain the class (or struct) name.
May or may not contain fields and methods.
There can be only one class per file.
Please do not put the closing bracket };. DRAKON Editor will do it for you.

=== options ===
C++ only. Currently, only one option is supported.
copying=no
Generates code that prohibits copying of the class instances.
copying=yes
Allows instances of the class to be copied.

Loop syntax

The *Loop start* icon is similar to the standard C *for* construct. It contains three expressions separated by semicolons. Here are two examples:



```
i = 0; i < length; i++
```

```
iterator = collection.begin();
iterator != collection.end();
++iterator
```

Please note that the loop variable should be declared at the beginning of the diagram.

Examples

Parameters item	Generated method header
public	void FunctionName() {
public function	void FunctionName() {
static function	static void FunctionName {
private method	private: void MethodName()
public static method int left int right returns const String*	public: static const String* MethodName(int left, int right) {
public virtual method returns int	public: virtual int MethodName() {
protected ctr String* foo	protected: ClassName(String* foo) {
dtr	public: virtual ~ClassName() {
protected virtual const abstract returns int	protected: virtual int MethodName() const = 0;