



Highlighting Function Names in LLDB Backtraces

Michael Buch

(lldb) bt

* thread #1, queue = 'com.apple.main-thread', stop reason = step in

* frame #0: 0x0000000100003444 a.out`boost::fusion::cons<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char

frame #1: 0x00000001000033c0 a.out`boost::fusion::cons<boost::spirit::detail::as_meta_element<boost::spirit::tag::char_code<boost:

00000010000b161) at make_cons.hpp:65:16
frame #2: 0x0000000100003340 a.out`boost::spirit::detail::make_terminal_impl<boost::proto::exprns_::expr<boost::proto::tags_::tag

operator()(this=0x000000016fdfe0ff, expr=0x000000010000b161, (null)=0x000000016fdfe0ee, data=0x000000016fdfe0fe) const at make_compone

frame #3: 0x00000001000032f8 a.out`boost::proto::detail::apply_transform<boost::proto::switch_<boost::spirit::meta_compiler<boost:

:spirit::char_encoding::standard>>, 0l> const& const&, mpl_::void_ const&, boost::spirit::unused_type& const&)>::result_type boost::pr

oost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l> cons

frame #4: 0x00000001000032e4 a.out`boost::spirit::result_of::compile<boost::spirit::qi::domain, boost::proto::exprns_::expr<boost:

mpiler<boost::spirit::qi::domain>::compile<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::ter

0000016fdfe0fd) at meta_compiler.hpp:243:24
frame #5: 0x00000001000031dc a.out`boost::spirit::result_of::compile<boost::spirit::qi::domain, boost::proto::exprns_::expr<boost:

ost::spirit::qi::domain, boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::t

frame #6: 0x0000000100003108 a.out`bool boost::spirit::qi::detail::phrase_parse_impl<boost::proto::exprns_::expr<boost::proto::tag

oost::proto::exprns_::basic_expr<boost::proto::tags_::tag::assign, boost::proto::argsns_::list2<boost::phoenix::actor<boost::proto::e

boost::proto::tags_::tag::dereference, boost::proto::argsns_::list1<boost::proto::exprns_::expr<boost::proto::tags_::tag::shift_righ

ns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::pr

oenix::actor<boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&>, 2l> const&>, 1l> const&>, 2l>, void>::call<std::_1::__wrap_iter

6fdfe340, last=(item = '\0'), expr=0x000000016fdfe318, skipper=0x000000010000b161, post_skip=postskip) at parse.hpp:82:43
frame #7: 0x0000000100001990 a.out`bool boost::spirit::qi::phrase_parse<std::_1::__wrap_iter<char const*>, boost::proto::exprns_:

boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::assign, boost::proto::argsns_::list2<boost::phoenix

::proto::exprns_::expr<boost::proto::tags_::tag::dereference, boost::proto::argsns_::list1<boost::proto::exprns_::expr<boost::proto::

ipt, boost::proto::argsns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns

<int>>, 0l>>, boost::phoenix::actor<boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&>, 2l> const&>, 1l> const&>, 2l>, boost::pro

0'), expr=0x000000016fdfe318, skipper=0x000000010000b161, post_skip=postskip) at parse.hpp:120:16
frame #8: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
frame #9: 0x0000000100001aa4 a.out`main at boost.cpp:32:22
frame #10: 0x0000000184785d54 dyld`start + 7184

(lldb) █

```
(lldb) f
```

```
frame #4: 0x000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
```

```
(lldb) f
frame #4: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
```

```
(lldb) settings show frame-format
frame-format (format-string) = "frame #${frame.index}: ${ansi.fg.cyan}${frame.pc}${ansi.normal}{ ${module.file.basename}{\`${function.name-with-args}{${frame.no-debug}${function.pc-offset}}}{ at ${ansi.fg.cyan}${line.file.basename}${ansi.normal}:${ansi.fg.yellow}${line.number}${ansi.normal}{:${ansi.fg.yellow}${line.column}${ansi.normal}}}{${function.is-optimized} [opt]}${function.is-inlined} [inlined]}${frame.is-artificial} [artificial]}\n"
```

```
(lldb) f
```

```
frame #4: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
```

```
(lldb) settings show frame-format
```

```
frame-format (format-string) = "frame #${frame.index}: ${ansi.fg.cyan}${frame.pc}${ansi.normal}{ ${module.file.basename}{\ ${function.name-with-args}{${frame.no-debug}${function.pc-offset}}}{ at ${ansi.fg.cyan}${line.file.basename}${ansi.normal}:${ansi.fg.yellow}${line.number}${ansi.normal}{:${ansi.fg.yellow}${line.column}${ansi.normal}}}{${function.is-optimized} [opt]}{${function.is-inlined} [inlined]}{${frame.is-artificial} [artificial]}\n"
```

```
(lldb) f
frame #4: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
```

```
(lldb) settings show frame-format
frame-format (format-string) = "frame #${frame.index}: ${ansi.fg.cyan}${frame.pc}${ansi.normal}{ ${module.file.basename}{\`${function.name-with-args}${frame.no-debug}${function.pc-offset}}}{ at ${ansi.fg.cyan}${line.file.basename}${ansi.normal}:${ansi.fg.yellow}${line.number}${ansi.normal}{:${ansi.fg.yellow}${line.column}${ansi.normal}}}{${function.is-optimized} [opt]}${function.is-inlined} [inlined]}${frame.is-artificial} [artificial]}\n"
```

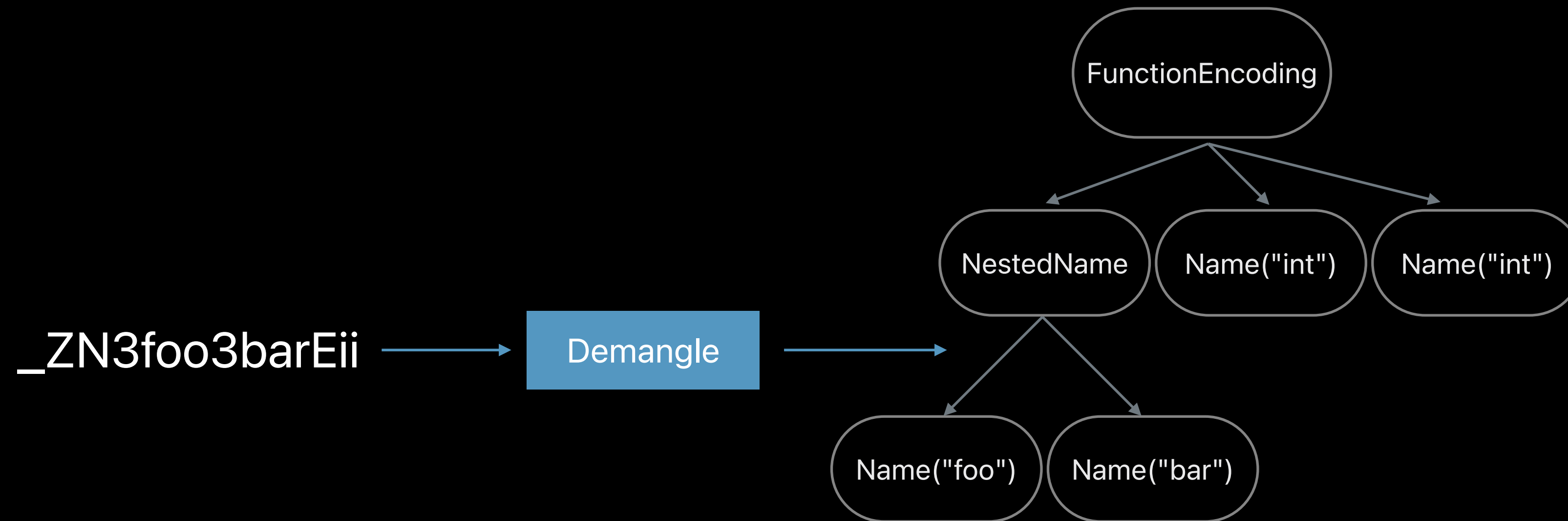
```
(lldb) f
frame #4: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
```

```
(lldb) settings show frame-format
frame-format (format-string) = "frame #${frame.index}: ${ansi.fg.cyan}${frame.pc}${ansi.normal}{ ${module.file.basename}{\`${function.name-with-args}${frame.no-debug}${function.pc-offset}}}{ at ${ansi.fg.cyan}${line.file.basename}${ansi.normal}:${ansi.fg.yellow}${line.number}${ansi.normal}{:${ansi.fg.yellow}${line.column}${ansi.normal}}}{${function.is-optimized} [opt]}${function.is-inlined} [inlined]}${frame.is-artificial} [artificial]}\n"
```

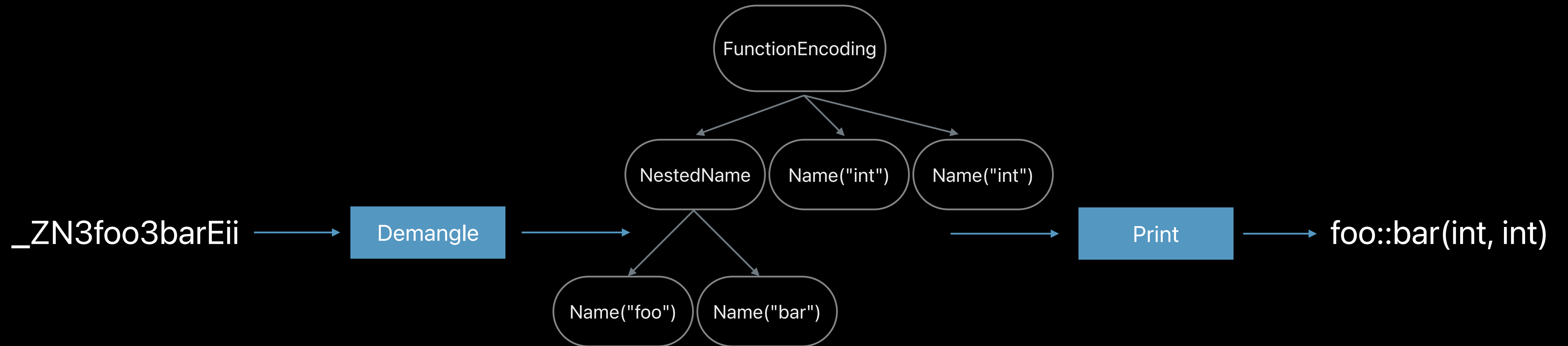
```
$ nm boost.o | grep parse_sum
0000000000000aa8 T __Z9parse_sumRKNS_3__112basic_stringIcNS_11char_traitsIcEENS_9allocatorIcEEEE
```

```
$ nm boost.o | grep parse_sum | xcrun llvm-cxxfilt
0000000000000aa8 T parse_sum(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char>> const&)
```

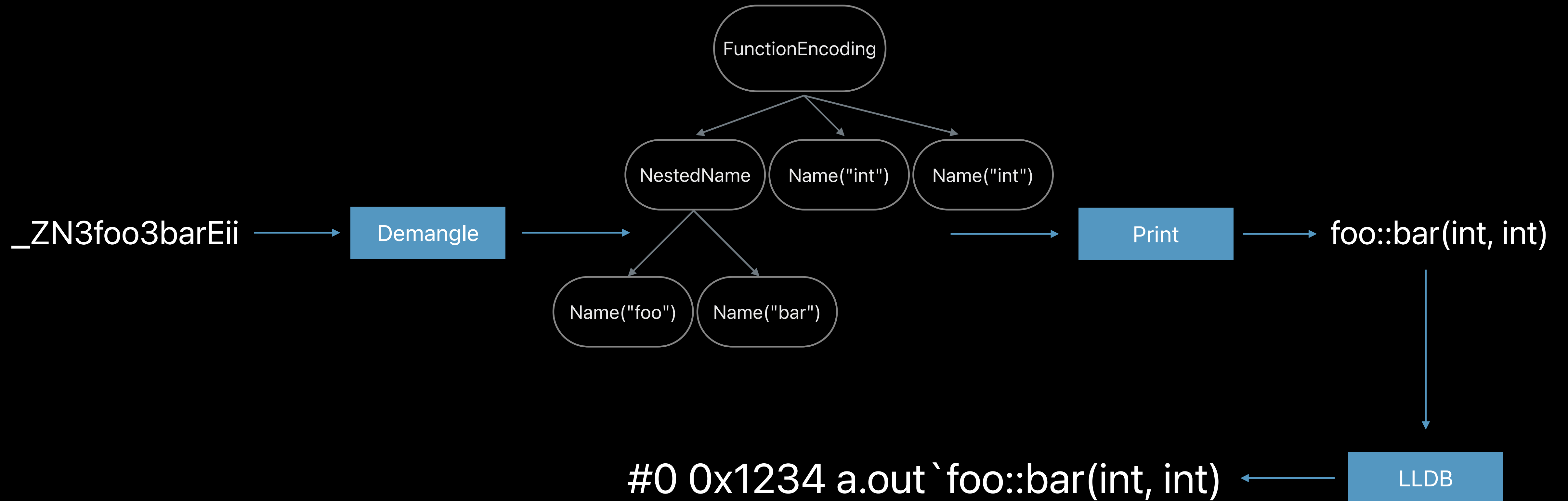
LLDB Demangler Integration



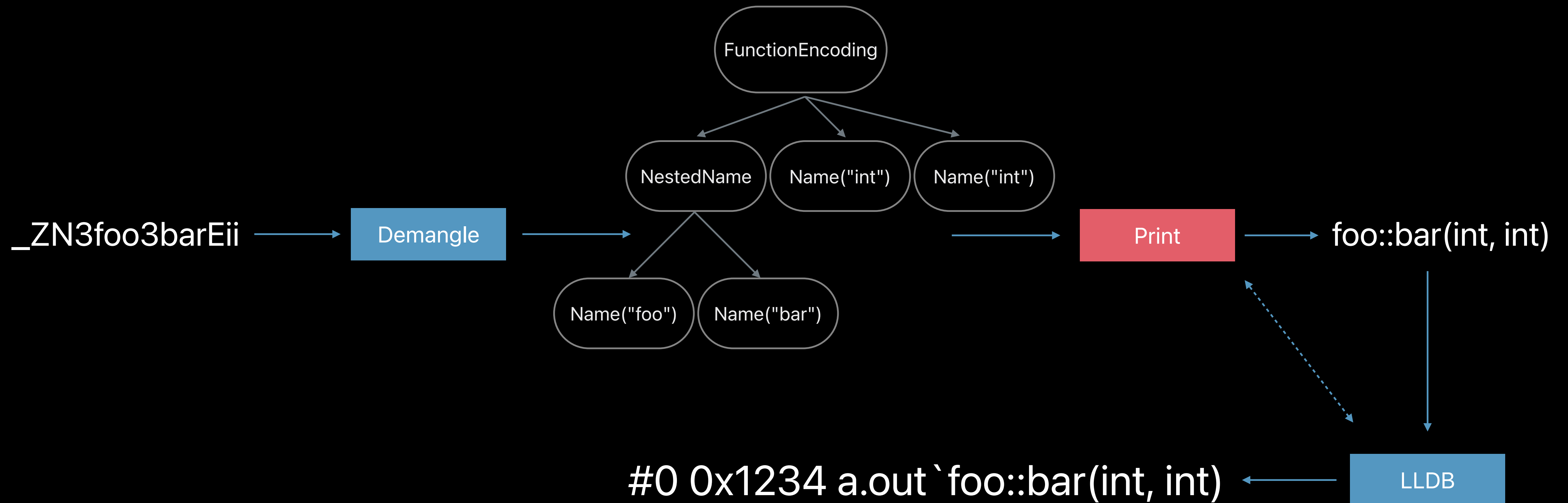
LLDB Demangler Integration

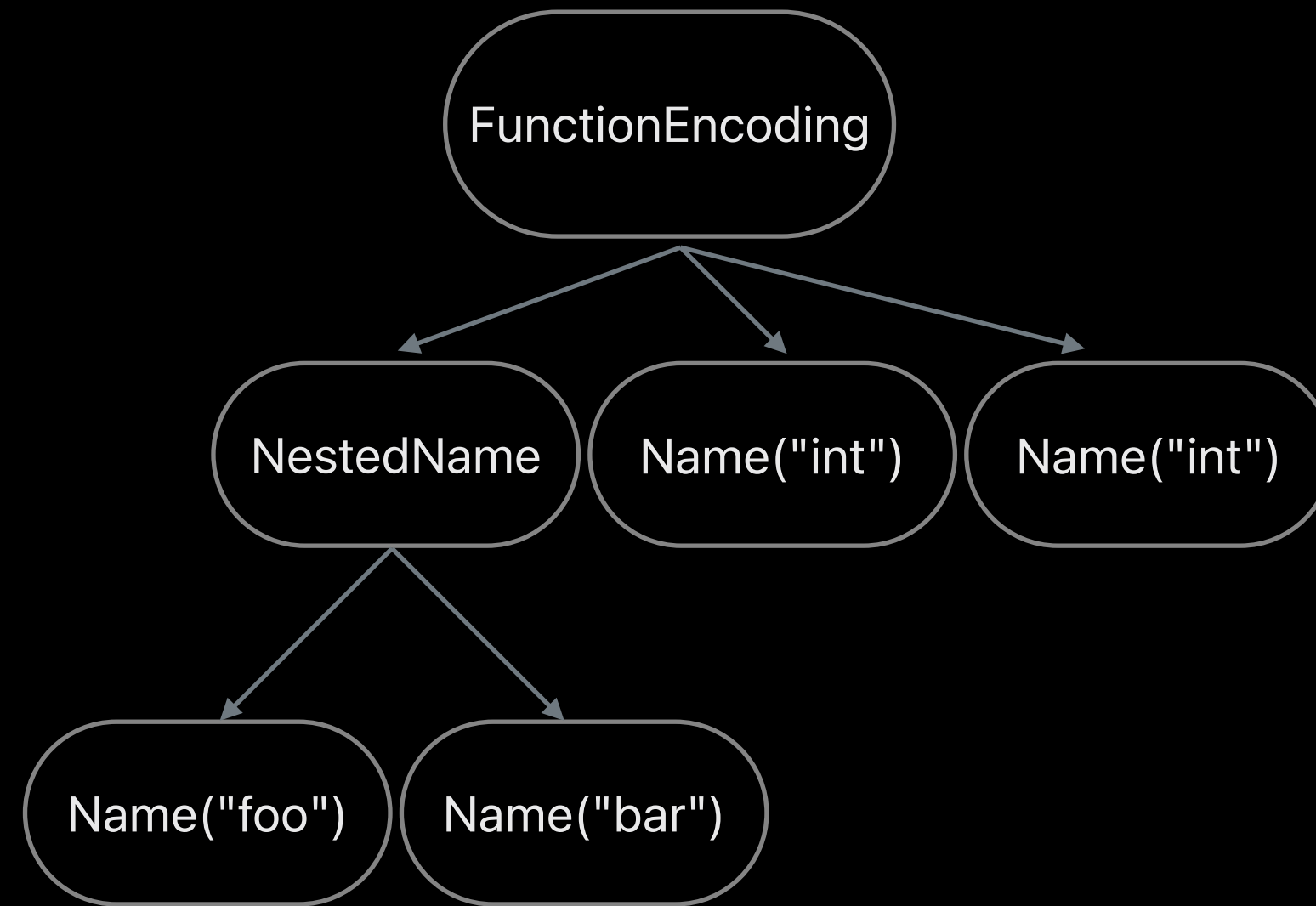


LLDB Demangler Integration

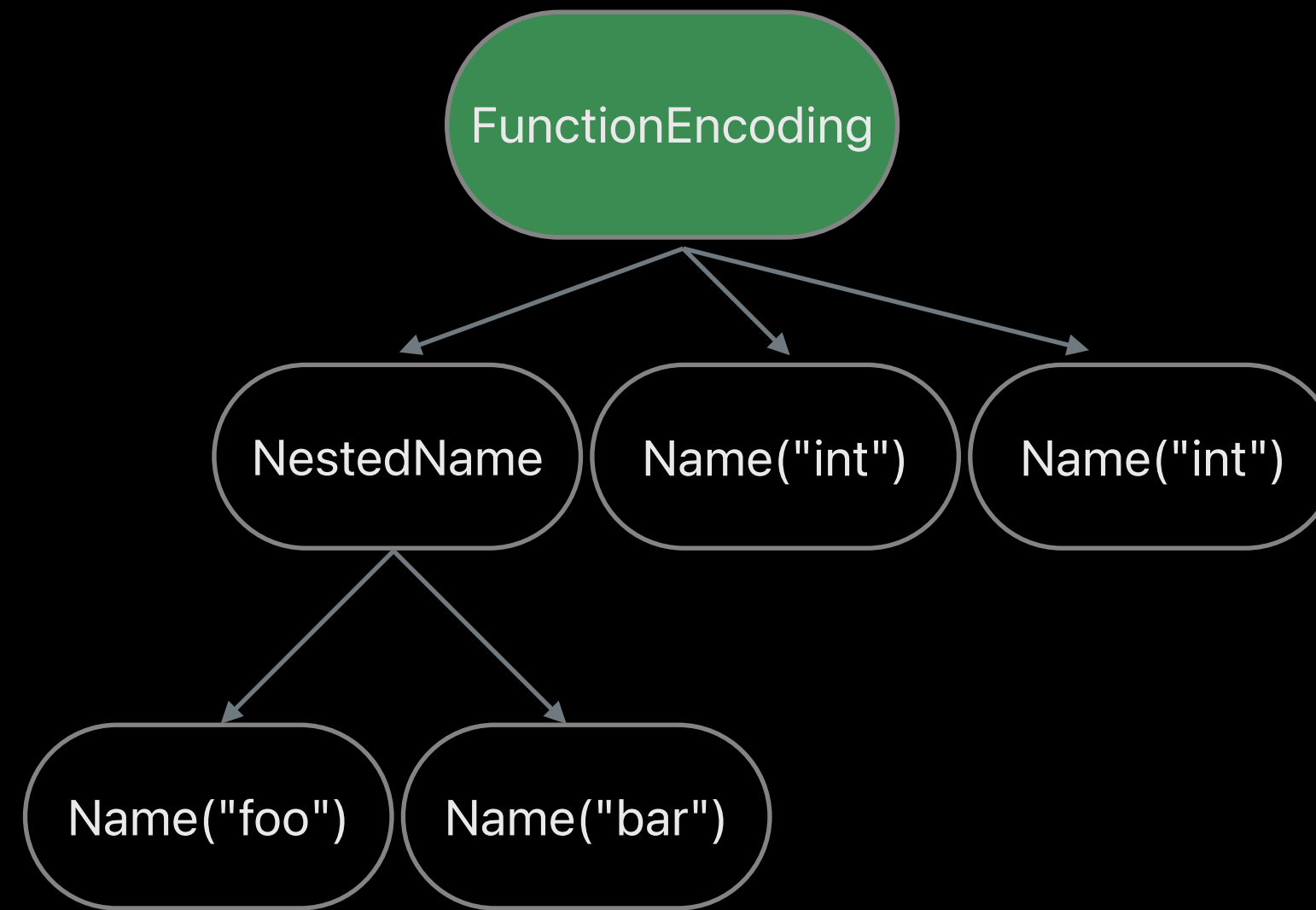


LLDB Demangler Integration



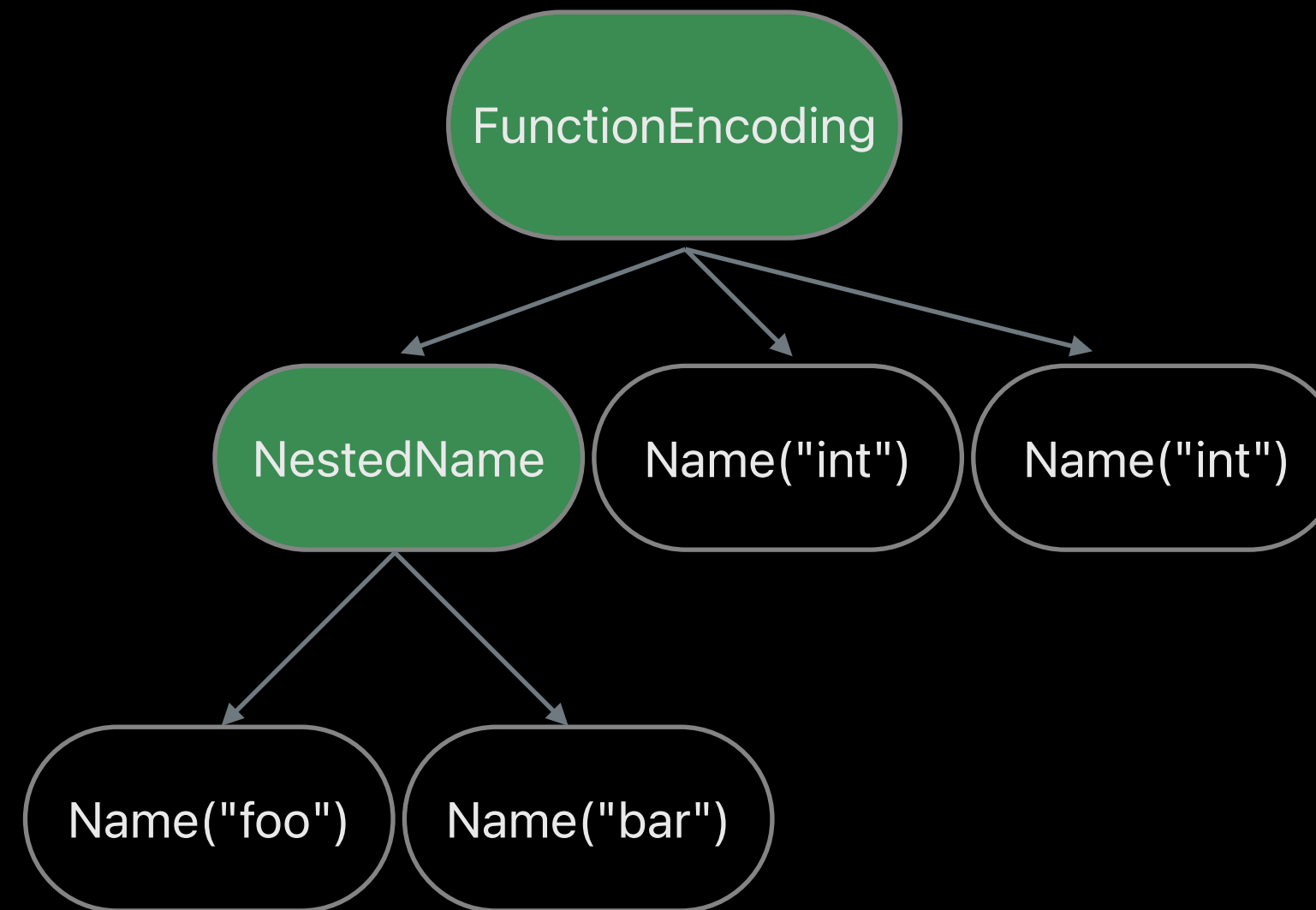


```
print(FunctionEncoding)
```



```
print(FunctionEncoding)
```

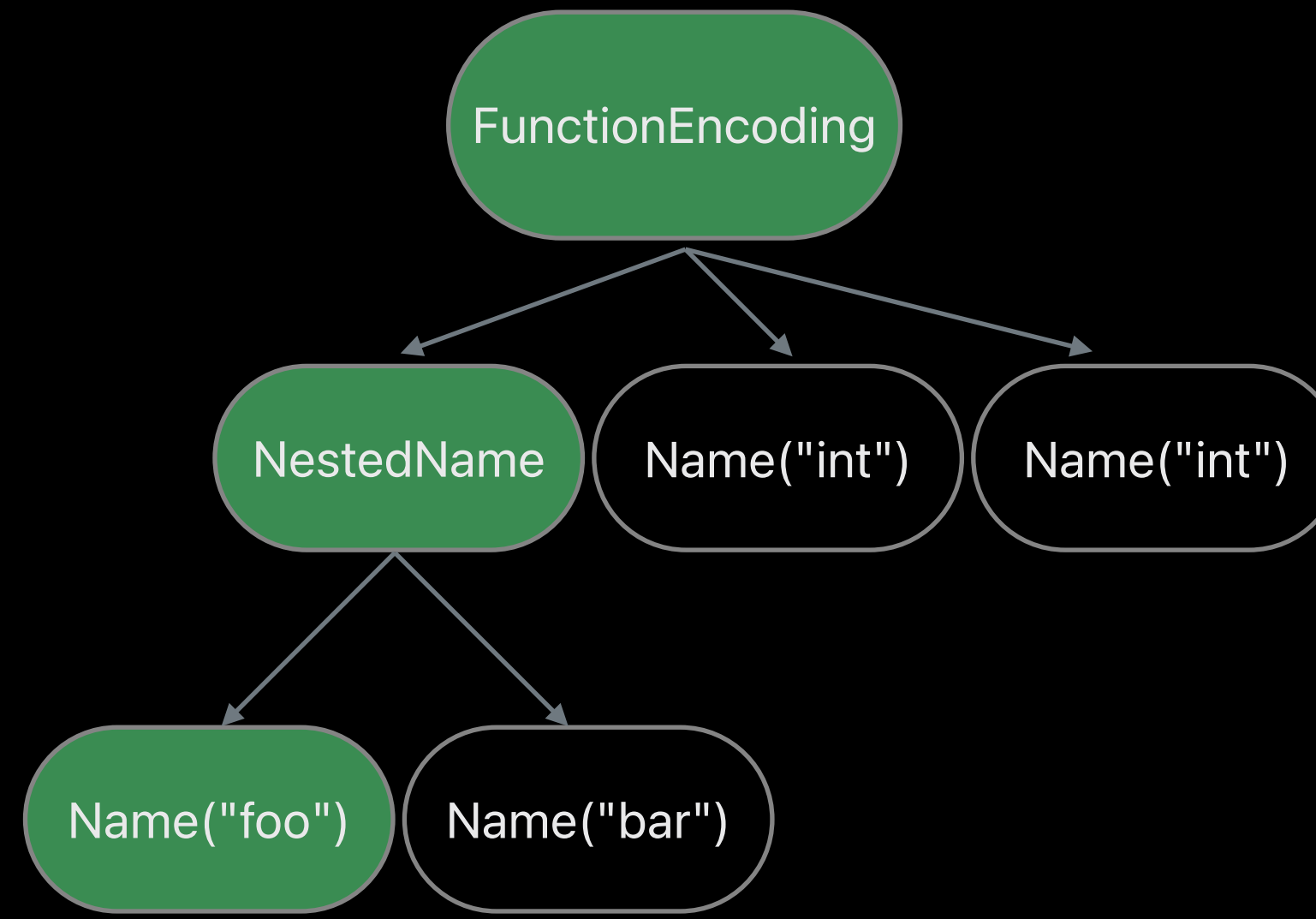
```
print(NestedName)
```



```
print(FunctionEncoding)
```

```
print(NestedName)
```

```
print(Name("foo"))
```



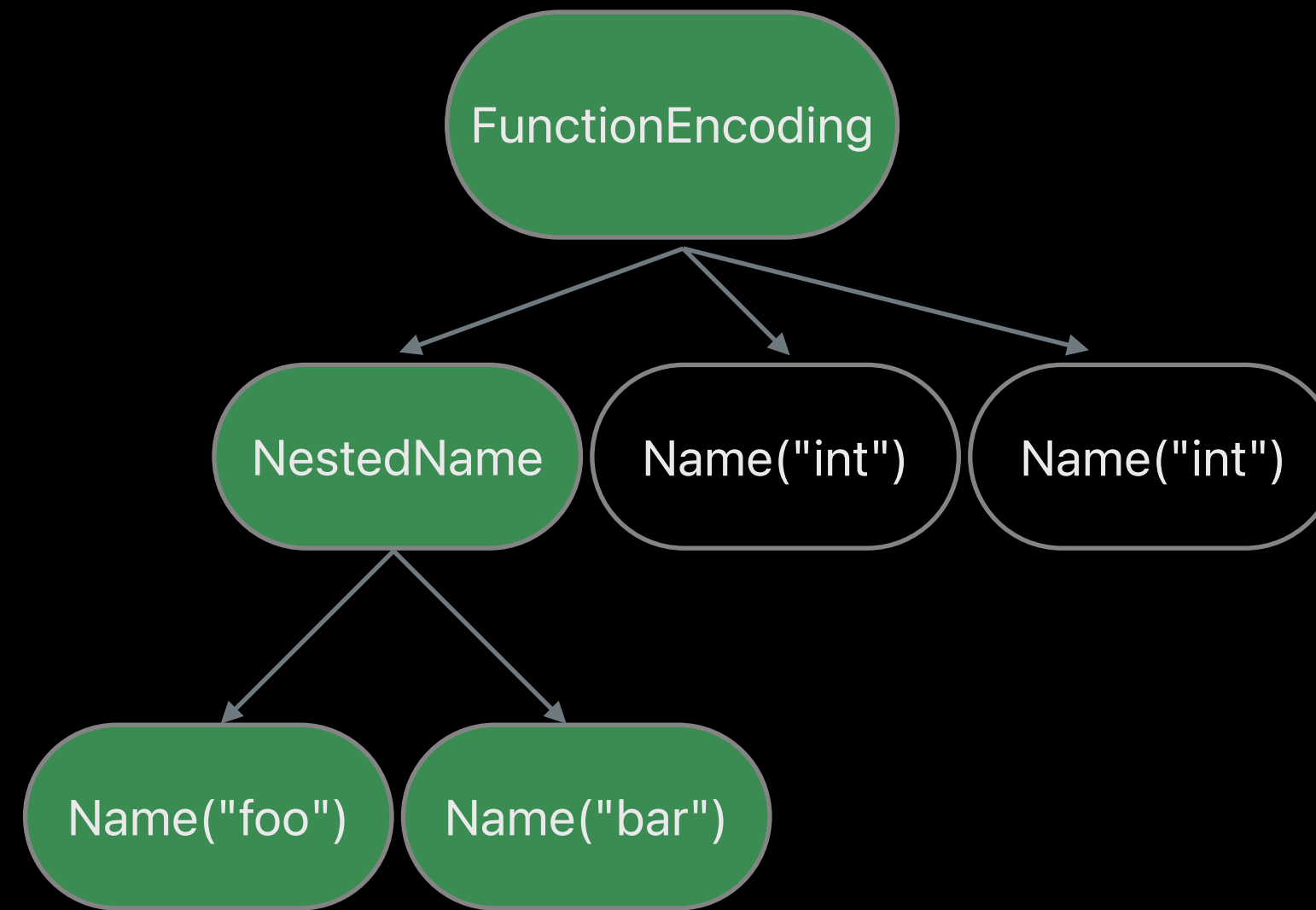
foo

```
print(FunctionEncoding)
```

```
print(NestedName)
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```



foo::bar

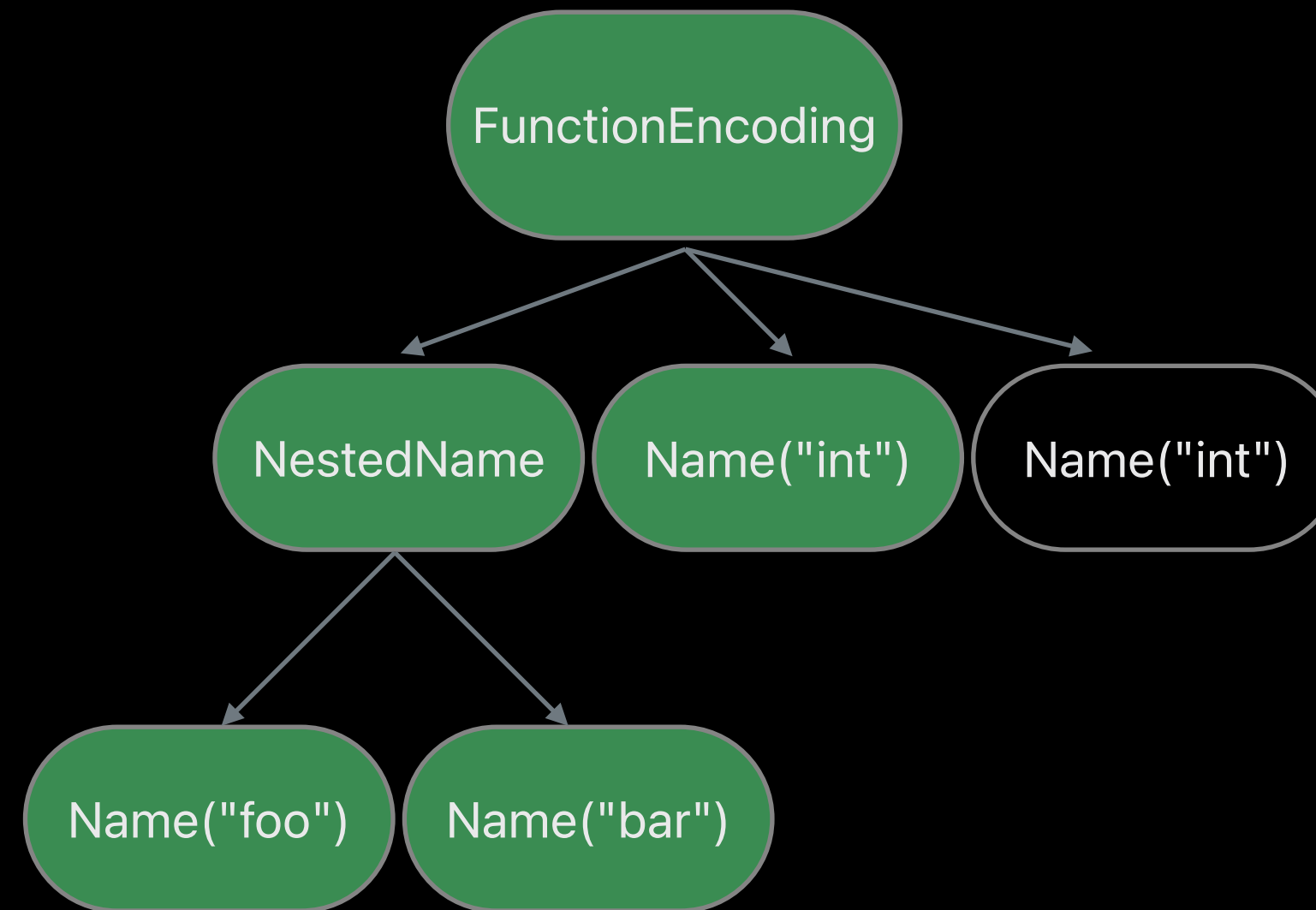
```
print(FunctionEncoding)
```

```
print(NestedName)
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```

```
print(Name("foo"))
```



```
foo::bar(int
```

```
print(FunctionEncoding)
```

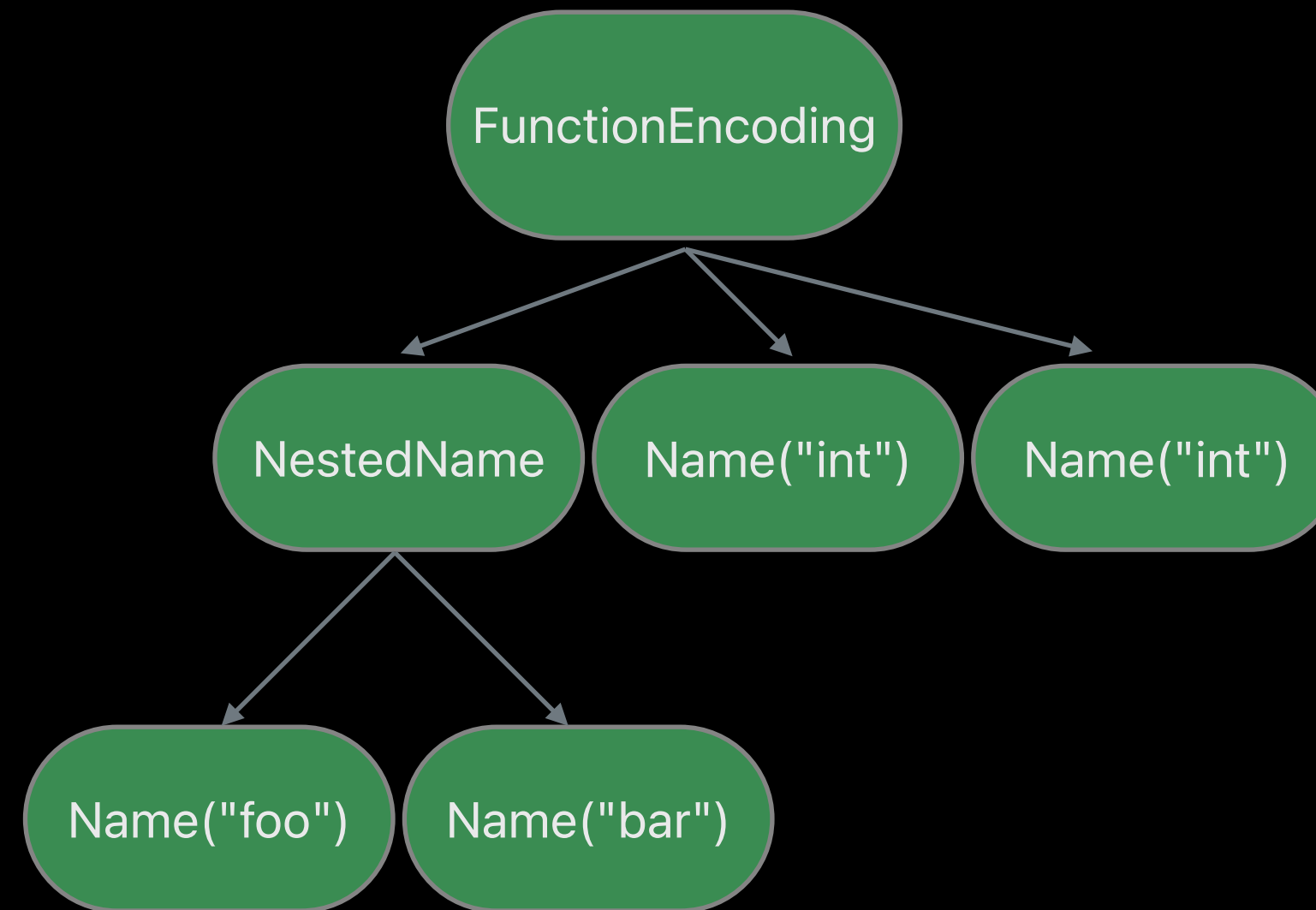
```
print(NestedName)
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```



```
foo::bar(int, int)
```

```
print(FunctionEncoding)
```

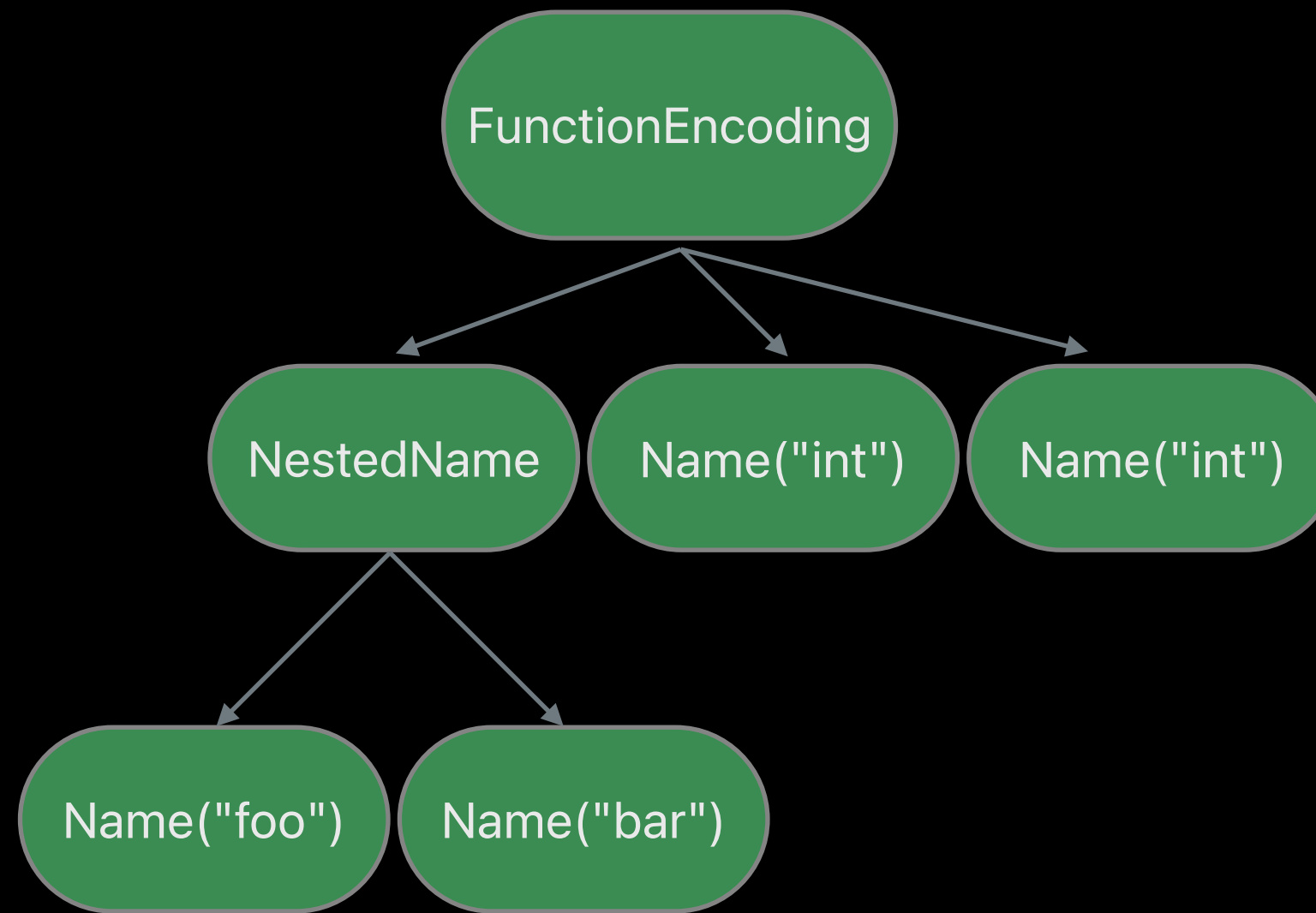
```
print(NestedName)
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```



```
foo::bar(int, int)
```

LLDB

```
print(FunctionEncoding)
```

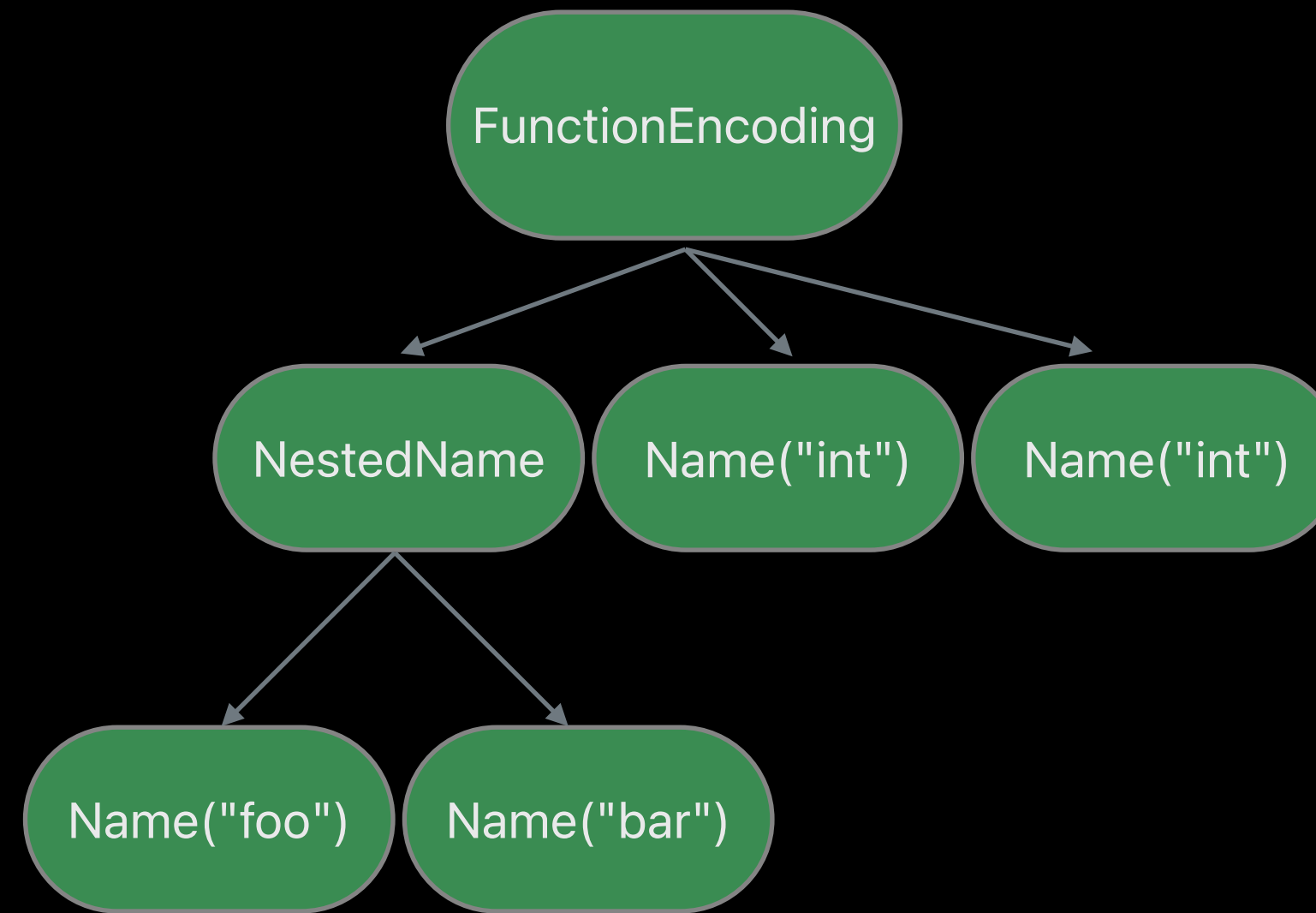
```
print(NestedName)
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```



```
foo::bar(int, int)
```

LLDB

```
NameInfo {  
    pair<u32, u32> Scope;  
    pair<u32, u32> BaseName;  
    pair<u32, u32> Arguments;  
};
```

```
print(FunctionEncoding)
```

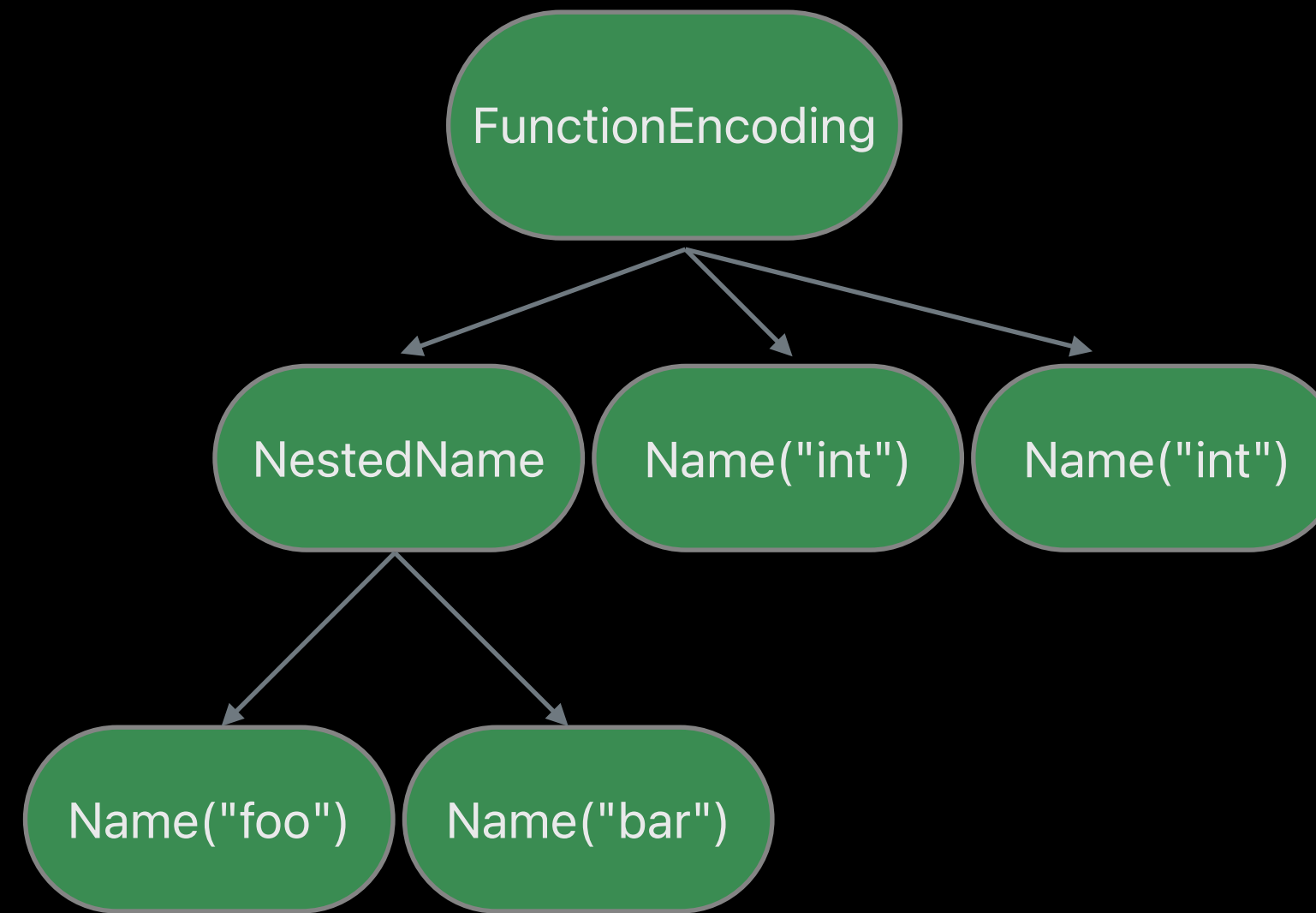
```
print(NestedName)
```

```
print(Name("foo"))
```

```
print(Name("bar"))
```

```
print(Name("foo"))
```

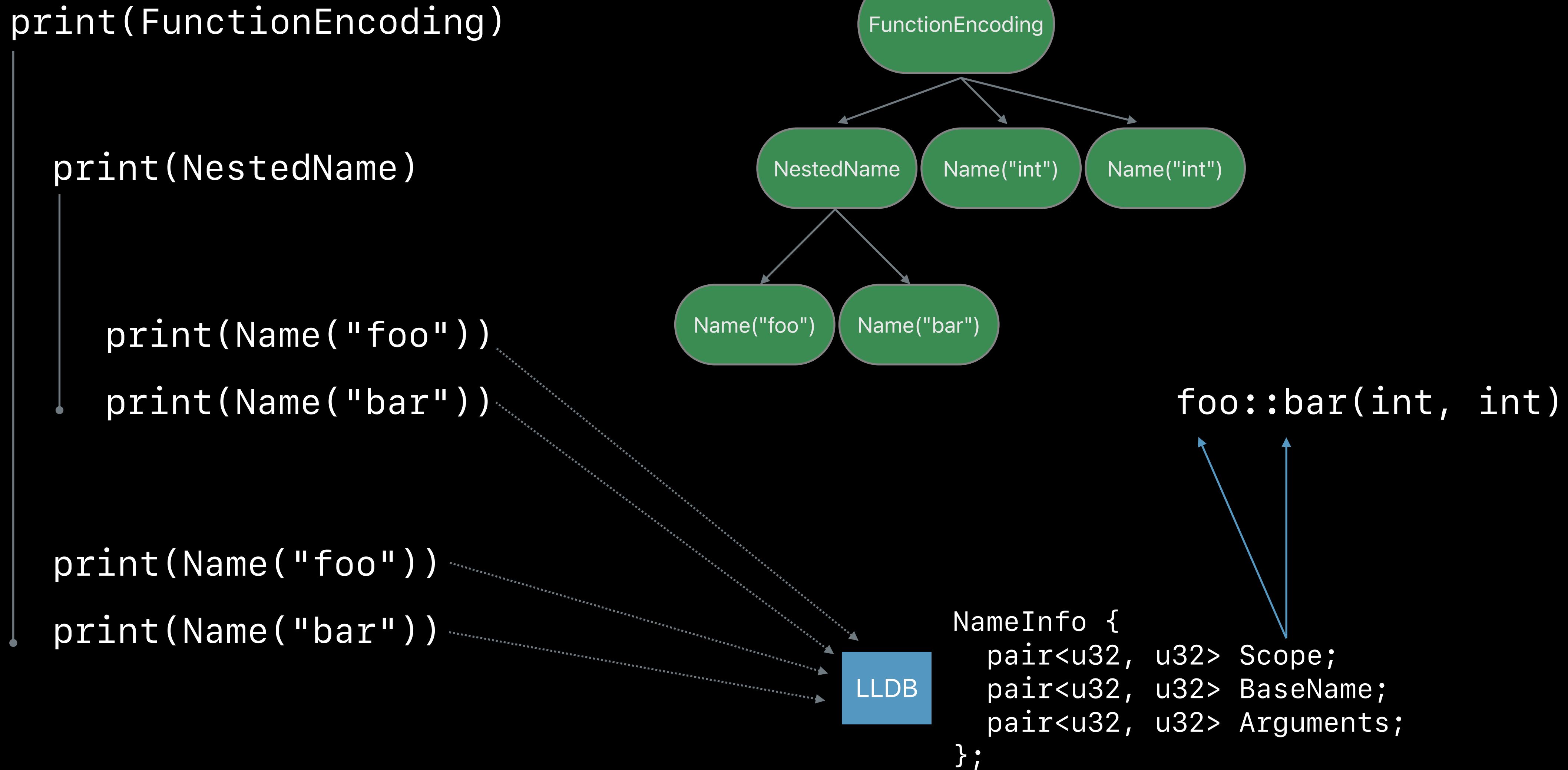
```
print(Name("bar"))
```



```
foo::bar(int, int)
```

LLDB

```
NameInfo {  
    pair<u32, u32> Scope;  
    pair<u32, u32> BaseName;  
    pair<u32, u32> Arguments;  
};
```



Function Name Formats

The function names displayed in backtraces/`frame info`/`thread info` are the demangled names of functions. On some platforms (like ones using Itanium the mangling scheme), LLDB supports decomposing these names into fine-grained components. These are currently:

- `${function.return-left}`
- `${function.prefix}`
- `${function.scope}`
- `${function.basename}`
- `${function.name-qualifiers}`
- `${function.template-arguments}`
- `${function.formatted-arguments}`
- `${function.qualifiers}`
- `${function.return-right}`
- `${function.suffix}`

```
(lldb) settings show plugin.cplusplus.display.function-name-format  
plugin.cplusplus.display.function-name-format (format-string) = "${function.return-left}${function.scope}${ansi.fg.yellow}${function.basename}${ansi.normal}${function.template-arguments}${function.formatted-arguments}${function.return-right}${function.qualifiers}${function.suffix}"
```

```
(lldb) settings show plugin.cplusplus.display.function-name-format
plugin.cplusplus.display.function-name-format (format-string) = "${function.return-left}${function.scope}${ansi.fg.yellow}${function.basename}${ansi.normal}${function.template-arguments}${function.formatted-arguments}${function.return-right}${function.qualifiers}${function.suffix}"
```



```
* thread #1, queue = 'com.apple.main-thread', stop reason = step out
* frame #0: 0x0000000100003480 a.out`boost::fusion::cons<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>, boost::fusion::nil_>::cons<this=0x0000000010000b161> at cons.hpp:60:28
    frame #1: 0x000000010000344c a.out`boost::fusion::cons<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>, boost::fusion::nil_>::cons<this=0x0000000010000b161> at cons.hpp:60:34
    frame #2: 0x00000001000033c0 a.out`boost::fusion::cons<boost::spirit::detail::as_meta_element<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>, boost::fusion::nil_> boost::spirit::detail::make_cons<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>(car=0x000000010000b161) at make_cons.hpp:65:16
    frame #3: 0x0000000100003340 a.out`boost::spirit::detail::make_terminal_impl<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l> const&, mpl_::void_ const&, boost::spirit::unused_type&, boost::spirit::qi::domain>::operator()<this=0x000000016fdfe09f, expr=0x000000016fdfe08e, data=0x000000016fdfe09e> const at make_component.hpp:86:17
    frame #4: 0x00000001000032f8 a.out`boost::proto::detail::apply_transform<boost::proto::switch_<boost::spirit::meta_compiler<boost::spirit::qi::domain>::cases, boost::proto::tag_of<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l> const&, boost::spirit::unused_type& const&>::result_type boost::proto::transform<boost::proto::switch_<boost::spirit::meta_compiler<boost::spirit::qi::domain>::cases, boost::proto::tag_of<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l> const&, mpl_::void_, boost::spirit::unused_type&>(this=0x000000016fdfe08f, e=0x000000010000b161, s=0x000000016fdfe08e, d=0x000000016fdfe09e) const at impl.hpp:255:9 [inlined]
    frame #5: 0x00000001000032e4 a.out`boost::spirit::result_of::compile<boost::spirit::qi::domain, boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l>, boost::spirit::unused_type, void>::type boost::spirit::detail::compiler<boost::spirit::qi::domain>::compile<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l>, boost::spirit::unused_type @ 0x000000016fdfe09e, (null)=mpl_::true_ @ 0x000000016fdfe09d) at meta_compiler.hpp:243:24
    frame #6: 0x00000001000031dc a.out`boost::spirit::result_of::compile<boost::spirit::qi::domain, boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l>, boost::spirit::unused_type, void>::type boost::spirit::detail::compiler<boost::spirit::qi::domain>::compile<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l>>(expr=0x000000010000b161) at meta_compiler.hpp:243:24
    frame #7: 0x0000000100003108 a.out`bool boost::spirit::qi::detail::phrase_parse_impl<boost::proto::exprns_::expr<boost::proto::tags_::tag::shift_right, boost::proto::argsns_::list2<boost::proto::tags_::tag::subscript, boost::proto::argsns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list2<boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&, boost::proto::exprns_::expr<boost::proto::tags_::tag::dereference, boost::proto::argsns_::list1<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list2<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<char const&>, 0l>, boost::proto::exprns_::expr<boost::proto::tags_::tag::plus_assign, boost::proto::argsns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&>, 2l> const&>, 1l> const&>, 2l>, void>::call<std::__1::__wrap_iter<char const*>, boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list1<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list2<boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&>, 2l> const&>, 1l> const&>, 2l>, boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l>>(first=0x000000016fdfe2e0, last=(item = '\\0'), expr=0x000000016fdfe2b8, skipper=0x000000010000b161, post_skip=postskip) at parse.hpp:82:43
    frame #8: 0x0000000100001990 a.out`bool boost::spirit::qi::phrase_parse<std::__1::__wrap_iter<char const*>, boost::proto::exprns_::expr<boost::proto::tags_::tag::shift_right, boost::proto::argsns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list2<boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&, boost::proto::exprns_::expr<boost::proto::tags_::tag::dereference, boost::proto::argsns_::list1<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::list2<boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<char const&>, 0l>, boost::proto::exprns_::expr<boost::proto::tags_::tag::plus_assign, boost::proto::argsns_::list2<boost::spirit::terminal<boost::spirit::tag::int_> const&, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::phoenix::actor<boost::proto::exprns_::basic_expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::reference_wrapper<int>>, 0l>>, boost::spirit::argument<0>>>, 2l>> const&>, 2l> const&>, 2l> const&>, 1l> const&>, 2l>, boost::proto::exprns_::expr<boost::proto::tags_::tag::terminal, boost::proto::argsns_::term<boost::spirit::tag::char_code<boost::spirit::tag::space, boost::spirit::char_encoding::standard>>, 0l>>(first=0x000000016fdfe2e0, last=(item = '\\0'), expr=0x000000016fdfe2b8, skipper=0x000000010000b161, post_skip=postskip) at parse.hpp:82:43
    frame #9: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost.cpp:16:15
```

```
$ cat ~/.lldbinit
settings set plugin.cplusplus.display.function-name-format "${ansi.fg.black}${ansi.bg.green}${function.basename}${ansi.normal}${function.formatted-arguments}"
```

```
$ cat ~/.lldbinit
settings set plugin.cplusplus.display.function-name-format "${ansi.fg.black}${ansi.bg.green}${function.basename}${ansi.normal}${function.formatted-arguments}"
```

```
$ cat ~/.lldbinit
settings set plugin.cplusplus.display.function-name-format "${ansi.fg.black}${ansi.bg.green}${function.basename}${ansi.normal}${function.formatted-arguments}"
```

```
$ cat ~/.lldbinit
settings set plugin.cplusplus.display.function-name-format "${ansi.fg.black}${ansi.bg.green}${function.basename}${ansi.normal}${function.formatted-arguments}"
```

```
$ cat ~/.lldbinit
settings set plugin.cplusplus.display.function-name-format "${ansi.fg.black}${ansi.bg.green}${function.basename}${ansi.normal}${function.formatted-arguments}"
```

```
(lldb) bt
```

```
* thread #1, queue = 'com.apple.main-thread', stop reason = step in
* frame #0: 0x0000000100003444 a.out`cons(this=0x000000016fdfe03d, in_car=0x0
frame #1: 0x00000001000033c0 a.out`make_cons(car=0x000000010000b161) at mak
frame #2: 0x0000000100003340 a.out`operator()(this=0x000000016fdfe09f, expr
frame #3: 0x00000001000032f8 a.out`operator()(this=0x000000016fdfe08f, e=0x
frame #4: 0x00000001000032e4 a.out`compile(expr=0x000000010000b161, modifie
frame #5: 0x00000001000031dc a.out`compile(expr=0x000000010000b161) at meta
frame #6: 0x0000000100003108 a.out`call(first=0x000000016fdfe2e0, last=(ite
frame #7: 0x0000000100001990 a.out`phrase_parse(first=0x000000016fdfe2e0, l
frame #8: 0x0000000100001728 a.out`parse_sum(input="10 + 20 + 30") at boost
```

Conclusion

- Shared repo enables
 - Low friction hooks into demangler
 - Testing LLDB by re-using LLVM demanglers existing test infrastructure

Conclusion

- Shared repo enables
 - Low friction hooks into demangler
 - Testing LLDB by re-using LLVM demanglers existing test infrastructure
- Future
 - Other languages (Swift already supported)
 - Expose to Xcode (and other IDEs)
 - Tool for interactive configuration of frame-format

