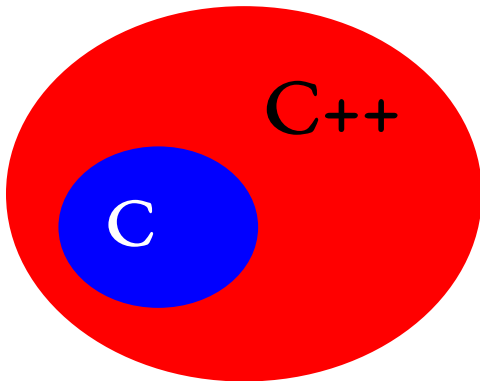


C $\not\subset$ C++

Frank Terbeck
`ft@bewatermyfriend.org`

January 9, 2020

$C \subset C++$ — Right!?



In Reality: $C \not\subset C++$

Quiz — Keywords

Code

```
int class = 2;  
int new = 3;  
bool mutable = false;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Keywords

Code

```
int class = 2;  
int new = 3;  
bool mutable = false;
```

C

All valid, and likely good names too.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Keywords

Code

```
int class = 2;  
int new = 3;  
bool mutable = false;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:8:5: error: expected primary-expression before 'int'  
test.cpp:9:9: error: expected unqualified-id before 'new'  
test.cpp:10:18: error: expected unqualified-id before '=' token
```

Code

```
struct FooBar {  
    int a;  
    int b;  
};  
/* ... */  
Foobar fb;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Code

```
struct FooBar {  
    int a;  
    int b;  
};  
/* ... */  
Foobar fb;
```

C

test.c:9:5: error: unknown type name 'FooBar'; use 'struct' keyword
to refer to the type

```
typedef struct FooBar { int a; int b; } FooBar;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — main()

Code

```
main(void)
{
    /* ... */
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — main()

Code

```
main(void)
{
    /* ... */
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C

test.c:1:1: warning: return type defaults to 'int' [-Wimplicit-int]

“Implicit int” should be an error since C99: -Werror-implicit-int

Quiz — main()

Code

```
main(void)
{
    /* ... */
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:1:10: warning: ISO C++ forbids declaration of 'main' with
               no type [-Wreturn-type]
```

Quiz — main() — Part II

Code

```
int  
main(void)  
{  
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — main() — Part II

Code

```
int  
main(void)  
{  
}
```

C

Valid since C99 (to match C++ sadly). Before:

```
test.c:4:1: warning: control reaches end of non-void function
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — `main()` — Part II

Code

```
int  
main(void)  
{  
}
```

C++

Valid C++. Returns 0. Only applies to `main()`.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto

Code

```
auto x = 2.25;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto

Code

```
auto x = 2.25;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C

```
test.c:4:10: warning: type defaults to 'int' in declaration of  
             'x' [-Wimplicit-int]
```

```
test.c:4:14: warning: implicit conversion from 'double' to 'int'  
             changes value from 2.25 to 2 [-Wliteral-conversion]
```


Quiz — auto

Code

```
auto x = 2.25;
```

C++

From C++11 x will be a double via type inference.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto, Part II

Code

```
auto int y = 5;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto, Part II

Code

```
auto int y = 5;
```

C

Valid C.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto, Part II

Questions

Code

```
auto int y = 5;
```

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:4:14: error: two or more data types in declaration of 'y'
```

```
test.cpp:4:5: warning: 'auto' storage class specifier is not permitted in C++11, and will not be supported in future releases [-Wauto-storage-class]
```

Quiz — auto, Part II

Code

```
register int z = 5;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto, Part II

Code

```
register int z = 5;
```

C

Valid C.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — auto, Part II

Questions

Code

```
register int z = 5;
```

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:4:5: error: ISO C++17 does not allow 'register' storage  
class specifier [-Wregister]
```

Quiz — VLA

Code

```
int x = 8;  
int a[x];
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — VLA

Code

```
int x = 8;  
int a[x];
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C

- Valid C (since C99): Variable Length Arrays.
- Optional since C11.
- Also: “USING VLAs IS ACTIVELY STUPID!” – Linus T.

Quiz — VLA

Code

```
int x = 8;  
int a[x];
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:5:10: warning: variable length arrays are a C99  
feature [-Wvla-extension]
```

Quiz — Initialisation

Code

```
typedef struct FooBar {  
    int a;  
    int b;  
} FooBar;  
FooBar fb = { .b = 1,  
              .a = 2 };
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Initialisation

Code

```
typedef struct FooBar {  
    int a;  
    int b;  
} FooBar;  
FooBar fb = { .b = 1,  
              .a = 2 };
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C

Valid C (since C99): Designated Initialisers

Quiz — Initialisation

Code

```
typedef struct FooBar {  
    int a;  
    int b;  
} FooBar;  
FooBar fb = { .b = 1,  
              .a = 2 };
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:6:41: error: designator order for field 'FooBar::a' does  
              not match declaration order in 'FooBar'
```

Quiz — Newer C Features

Code

```
_Atomic char f = 'f';  
_Bool g = true;
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Newer C Features

Code

```
_Atomic char f = 'f';  
_Bool g = true;
```

C++

```
test.cpp:4:5: error: ‘_Atomic’ was not declared in this scope  
test.cpp:5:5: error: ‘_Bool’ was not declared in this scope
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Newer C Features

Code

```
#define cbrt(x)
    _Generic((x),
        long double: cbrt1, \
        float: cbrtf        \
        default: cbrt,      \
    )(x)
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Newer C Features

Code

```
#define cbrt(x)
    _Generic((x),
        long double: cbrt1, \
        float: cbrtf        \
        default: cbrt,      \
    )(x)
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:8:5: warning: generic selections are a C11-specific
              feature [-Wc11-extensions]
```

Quiz — void Conversion

Code

```
int *ii;  
ii = malloc(sizeof(*ii));
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — void Conversion

Code

```
int *ii;  
ii = malloc(sizeof(*ii));
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

```
test.cpp:6:21: error: invalid conversion from 'void*' to 'int*' [-fpermissive]
```

Quiz — Character Literals

Code

```
size_t chlitsize = sizeof('%');
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Character Literals

Code

```
size_t chlitsize = sizeof('%');
```

C

In C, a character literal is an `int`.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Character Literals

Code

```
size_t chlitsize = sizeof('%');
```

C++

In C++, a character literal is a `char`.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Implicit Function Declaration

Code

```
int
main(void)
{
    /* No previous decl! */
    quux();
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Implicit Function Declaration

Code

```
int
main(void)
{
    /* No previous decl! */
    quux();
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C

test.c:4:5: warning: implicit declaration of function 'quux'

Should be an error since C99: -Werror-implicit-function-declaration

Quiz — Implicit Function Declaration

Code

```
int
main(void)
{
    /* No previous decl! */
    quux();
}
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

C++

test.cpp:4:5: error: 'quux' was not declared in this scope

Quiz — Huh?

Code

```
:%:include <stdio.h>
:%:include <stdlib.h>
int main(int argc, const char *argv<::>) <%
    while (argc-- > 1) <%
        printf("%s\n", argv<:argc:>);
    %>
    return EXIT_SUCCESS;
%>
```

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Quiz — Huh?

Code

```
%:include <stdio.h>
%:include <stdlib.h>
int main(int argc, const char *argv<::>) <%
    while (argc-- > 1) <%
        printf("%s\n", argv<:argc:>);
    %>
    return EXIT_SUCCESS;
%>
```

Valid C and C++! — Digraphs.

Questions

- Valid C?
- Valid C++?
- Both?
- Neither!?

Closing Thoughts

- These are not all differences!
- Almost everything is language version dependent.
- Almost everything is toolchain dependent.
- Idiomatic C is not idiomatic C++.
- Idiomatic C++ is probably not valid C.
- Interfaces are the solution for interoperation.
 - Using C from C++ is very easy.
 - Using C++ from C is probably not.
- Compiling C with a C++ compiler is not.

Thanks for your attention!