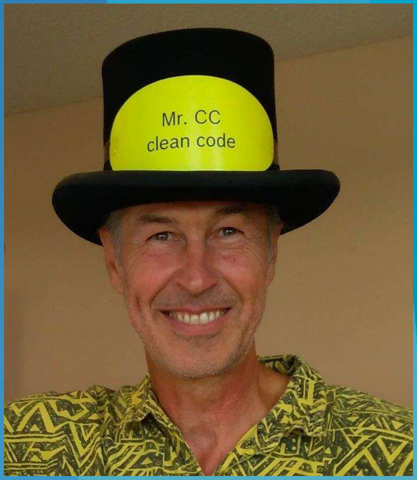
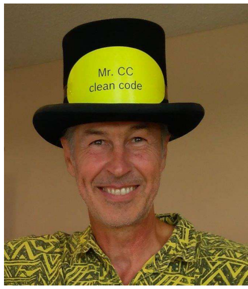
LIFE CYCLE SESSION 1 TOOLS AND ACTIVITIES



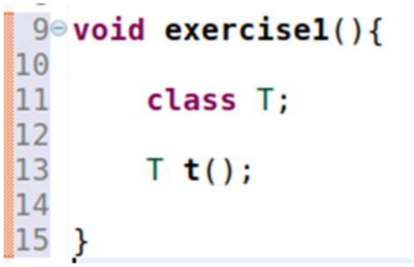
Mr. CC clean code
Gerd Hirsch (CC-AD/ESW1)

BOSCH

C++ Basics: Lifecycle
Session 1 Solution 1 and 2



- Is this compilable, and – if not, why?
  - ☒ Yes
  - □ No, cause …
- The most vexing parse C++
  - https://en.wikipedia.org/wiki/Most_vexing_parse
  - **class T;** is a declaration of a class named T
  - **T t();** is interpreted as a declaration of a function named **t** returning an object of **T**
    - Everything that looks like a function declaration will be interpreted as one
  - Declarations are allowed in functions



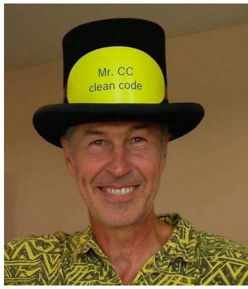
```cpp
void exercise1(){

    class T;

    T t();

}
```

C++ Basics: Lifecycle
Session 1 Solution 1 and 2

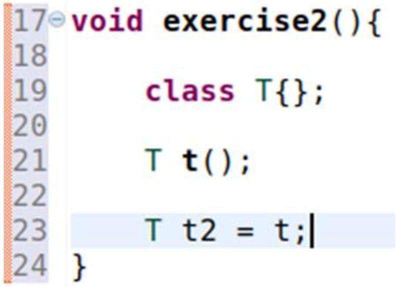


- Is this compilable, and – if not, why?

□ ▶ Yes

☒ ▶ No, cause **T t2 = t;** is the definition of an object t2, which requires

  - the definition of class T{};
  - with an constructor expecting a function as an argument: **T(T()){}**
  - but there is no such constructor in class T

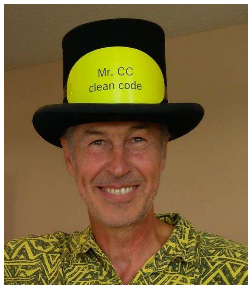


```
void exercise2(){

    class T{};

    T t();

    T t2 = t;
}
```

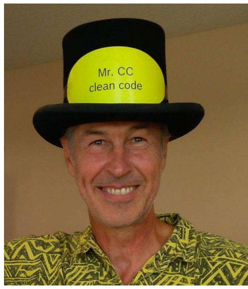


C++ Basics: Lifecycle
Session 1 Solution 2

- Compiler Message

```
../src/exercises.cpp:21:9: error:
conversion from ‘exercise2()::T()’ to non-scalar type ‘exercise2()::T’ requested
  T t2 = t;
       ^
```

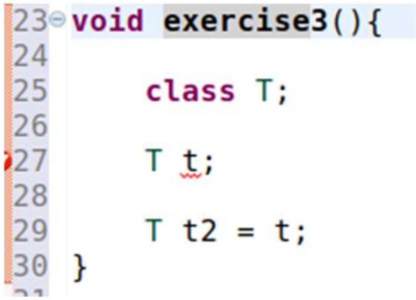
C++ Basics: Lifecycle
Session 1 Solution 3



- Is this compilable, and – if not, why?

□ ▶ Yes

☒ ▶ No, cause
T t; is the definition of an object of T and
T t2 = t; is also the definition of an object t2, which requires
the definition of class T{}; but there is only a class T; declaration



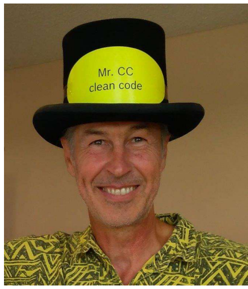
```
void exercise3(){

    class T;

    T t;

    T t2 = t;
}
```

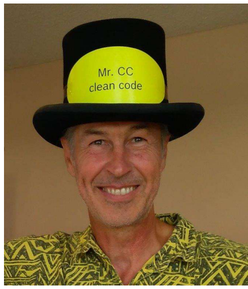
C++ Basics: Lifecycle
Session 1 Solution 3



- Compiler Message

```
../src/exercises.cpp:29:4: error:
aggregate ‘exercise3()::T t’ has incomplete type and cannot be defined
  T t;
    ^
../src/exercises.cpp:31:4: error:
variable ‘exercise3()::T t2’ has initializer but incomplete type
  T t2 = t;
```

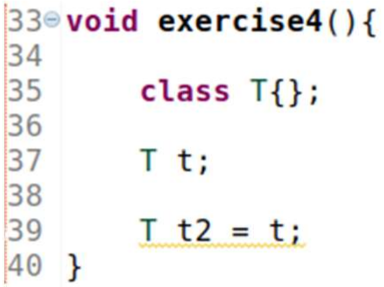
C++ Basics: Lifecycle
Session 1 Solution 4



- Is this compilable, and – if not, why?

☒ ▶ Yes, cause

  - **T t;** is the definition of an object of class T named t, initialized by the compiler synthesized default constructor
  - **T t2 = t;** is the definition of another object t2 initialized by the compiler synthesized copy constructor
  - which requires the definition of **class T{};**



```cpp
void exercise4(){

    class T{};

    T t;

    T t2 = t;
}
```