The definition of `TCB_SIZE` is shown below. `StaticTask_t` is the FreeRTOS TCB, and it's defined in `FreeRTOS.h`.

```
#define TCB_SIZE (sizeof(StaticTask_t))
```

`TCB_SIZE` is defined in `RTOS_Dependencies.h`. That header-file was created by the book's author, along with the other code he developed for Chapter 14.

`RTOS_Dependencies.h` is used for the CMSIS-RTOS dependencies on the underlying RTOS (here, it's FreeRTOS). A header-file like this is useful when porting the application program to another RTOS. The header-file can be copied, and updated with the specific values for the other RTOS.

- **Bug in the code comments** (`main_taskCreation_CMSIS_RTOSV2.c`)
  - Part of this code-comment is incorrect:

```
/**
 * since we don't want this code to have direct dependencies on FreeRTOS.h,
 * we can't use sizeof(StaticTask_t) directly - instead TCB_SIZE is declared in
 * Nucleo_F767ZI_Init.h and defined in Nucleo_F767ZI_Init.c
**/
uint8_t RedTask_TCB[TCB_SIZE];
```

  - The code-comment should say:

```
/**
 * The FreeRTOS TCB is StaticTask_t.
 * Since we don't want the present program to have direct dependencies on
   FreeRTOS.h, we can't use sizeof(StaticTask_t) directly - instead TCB_SIZE is
   used. TCB_SIZE is a macro, and it is defined in RTOS_Dependencies.h
**/
```

- **Clarification**, POSIX installation (pg 381)
  - The steps shown describe how to install POSIX with the example program. The text incorrectly implies that these steps will need to be performed by the reader. The example program comes with POSIX already installed.

- **Clarification,** "least common denominator" (pg 388)
  - The sentence below seems to be misworded:

When writing code that is intended to be run across multiple targets, a least common denominator approach will need to be taken – be sure to only use the smallest number of features commonly available across all target platforms.

  - This clause appears to be misworded, "*use the smallest number of features commonly available across all target platforms*". It is intended to say, "*use features commonly available across all target platforms* ".
  - In general, I think the phrase "*least common denominator*" is often a misnomer, as a metaphor. The metaphor is based on a mathematical concept, but the correlation to it is often unclear. Here, it seems that "greatest common denominator" is intended, i.e., the largest set of features that are available on all of the targets.