HW5-1

Explanation: How interval_1ms value is determined
RealViewPB uses a **1 MHz timer clock** as shown in the lab material.
Lab05_timer

That means the timer increments **1,000,000 times per second**, or **1 tick = 1 microsecond (1µs)**.
The timer reload value is calculated using the formula given in the slide:
TimerXLoad = (Interval × TIMCLK_FREQ) / (TIMCLKENx_DIV × PRESCALE_DIV)
So:
TimerXLoad = 0.001s × 1,000,000 Hz / (1 × 1)
        = 1,000 ticks
The code in Slide 9 does exactly that by computing:

```
uint32_t interval_1ms = TIMER_1MZ_INTERVAL / 1000;   // = 1,000,000 / 1000 = 1000
Timer->timerxload = interval_1ms;
```

So the timer reload value becomes **1000**, meaning the timer counts down 1000 clock cycles before generating an interrupt — which equals **1 millisecond**.

HW5-2

Why the line Timer->timerxintclr = 1; is necessary
Inside the timer interrupt handler:

```
static void interrupt_handler(void)
{
   sInternal_1ms_counter++;
   Timer->timerxintclr = 1;   // <-- IMPORTANT
}
```

This line is required because writing **1** to TIMERXINTCLR register **clears the interrupt flag** in the timer hardware.
Without this line:

- The interrupt status bit remains set,
- The timer will repeatedly trigger the interrupt **immediately**,
- The CPU will stay stuck inside the interrupt handler forever,
- The system would never return to normal program execution.

So this write operation tells the timer:
The interrupt has been serviced. Clear the flag so next interrupt occurs only after the next 1 ms countdown.