CS6640 Week4.a

1. Timer interrupt handler

```
void handler() {
    CRITICAL("Got a timer interrupt!");
    // (4) reset timer

}

int main() {
    CRITICAL("This is a simple timer example");

    // (1) register handler() as interrupt handler


    // (2) set a timer


    // (3) enable timer interrupt


    while(1);
}
```


2. Background: RISC-V assembly II

Assembler instructions with C expression operands:

asm(Template : OutputOperands : InputOperands)

a) Template: a string that is the template for the assembler code.

```
asm("mret");
```

b) OutputOperands: the C variables modified by the instructions in the Template.

```
void *sp;
asm("mv %0, sp" : "=r"(sp));
```

c) InputOperands: C expressions read by the instructions in the Template.

```
int mie;
asm("csrr %0, mie" : "=r"(mie));
asm("csrw mie, %0" ::"r"(mie | 0x80));
```