

COMP500 / ENSE501: Week 11 – Exercise:

EXERCISE NAME: *XOR Swap*

Given the following program source code:

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  void example_swap(void);
5  void example_xor_swap(void);
6
7  int main(void)
8  {
9      example_swap();
10     printf("\n");
11     example_xor_swap();
12
13     return 0;
14 }
15
16 void example_swap(void)
17 {
18     int a = 135;
19     int b = 246;
20
21     printf("BEFORE: a is %d, b is %d\n", a, b);
22
23     int temp = a;
24     a = b;
25     b = temp;
26
27     printf("AFTER:  a is %d, b is %d\n", a, b);
28 }
29
30 void example_xor_swap(void)
31 {
32     int c = 123;
33     int d = 456;
34
35     printf("BEFORE: c is %d, d is %d\n", c, d);
36
37     c = c ^ d;
38     d = c ^ d;
39     c = c ^ d;
40
41     printf("AFTER:  c is %d, d is %d\n", c, d);
42 }
```

Notice how the **example_swap** function uses a **temp** variable to conduct the swapping of **a** and **b** – this is the conventional way of swapping two values – whereas the **example_xor_swap** function avoids using a temporary. Trace the execution of this program, and keep track of the binary representation of each variable **a**, **b**, **c**, **d** and **temp**. Record your trace in your Reporting Journal.