CS5243 Advanced UNIX Programming
Assignment 5 (4 pts)
Group 4

(1)
Yes, it is correct. Even if we revise the function f1 so that f1 returns the address of the value.
The first line prints the value and address of val inside the f1 function. The second line prints the value and address of val from the f1 function. The values and addresses of both lines are the same.

```
freebsd@generic:~/Advanced-UNIX-Programming_Student/assignment5 % ./assignment5
Value 5 is at 0x80617ffc in function f1
Value 5 is at 0x80617ffc
freebsd@generic:~/Advanced-UNIX-Programming_Student/assignment5 %
```

The reason is that GCC compiler will optimize the memory of the variable to ensure that there is no error.

Sample output:

```
freebsd@generic:~/Advanced-UNIX-Programming_Student/assignment5 % ./assignment5
Value 0 is at 0x8069e0cc
```

(2)
The screenshot of the code

```c
#include <stdio.h>
int* f1(int val){

    int num = 0;
    int*ptr = &num;
    // printf("Value %d is at %p in function f1\n", *ptr, ptr);

    if(num == 0){
        /* text had val == 0, but val is not in scope */
        int val;
        val = 5;
        ptr = &val;
        // printf("Value %d is at %p in function f1\n", *ptr, ptr);
    }

    return ptr;
}
int main(void){

    int val = 5;
    int *ptr = f1(val);

    printf("Value %d is at %p\n", *ptr, ptr);

    return 0;
}
```