Assignment 5

(1) (1 pt) Answer the question above. You need to explain why it is correct/ incorrect in your report.

Answer: The given code is incorrect, since it references the automatic stack allocated integer val through a pointer after the automatic variable is no longer in existence. Automatic variables declared within a block are not available outside the block, as the stack frame is removed.

(2) Implement and modify the above C code to verify your answer.

a. **(1 pt)** Create a variable called val and set its value to "5" in the main function.

Answer: Variable `val` is created as an integer pointer. We allocate space on the heap to the pointer using `malloc` and assign the space to `5` here in the `main()` function:

```
int *val = malloc(sizeof(int));
*val = 5;
```

b. **(1 pt)** Get the address of val from the f1 function. (Getting the address in the main function directly is forbidden!)

Answer: We get the address of the memory allocated by malloc in `f1` as follows:

```
return &(*val);
```

We pass a pointer to `f1` since otherwise, the parameter passed to `f1` will contain a copy of the original data, making the address calculation incorrect. Then we store the result in a pointer `valp`:

```
int* valp = f1(val);
```

c. **(1 pt)** Print the value and the address of val in the main function as the sample output shows.

Answer: We use the `%p` format specifier to print the pointer address as per the spec.

```
printf("Value %i is at %p\n", *val, valp);
```