

1. What is wrong with the following code and how would you fix it?

```
int min;
    for (int i=0; i<10; ++i) {
        int value;
        scanf("%d", &value);
        if(i=0) min = value;
        else if(value<min) min = value;
    } // end of loop over input
printf("The smallest or the 10 values you entered was %d\n", min);
```

Before the loop starts, the min value is not initialized with any integer number. As the variable value is declared within the for loop; thus, after the for loop ends, variable value will be out of scope and deleted by the computer. Variable min refers to value within the for loop, now will refers to some unused part of memory out of the for loop.

The fix would be

2. What is wrong with the following code and how would you fix it?

```
int* pointerToMax(int a, int b) {
    if(a>b) return &a;
    else return &b;
}
```

3. What is wrong with the following code and how would you fix it?

```
int* ptr = NULL;
scanf("%d", ptr);
printf("You entered %d", &ptr)
```

4. Describe what each of the following declare:

- (a) int\* a;
- (b) const int b;
- (c) const int\* c;
- (d) int\* const d;
- (e) const int\* const e;

5. What is the difference between the \* and & operators?

& returns a pointer to a variable, while the \* does the opposite by dereferences the pointer and return the contents of the memory location it points to.