

Ethan Tuning  
2/2/2017  
CSCD240  
Lab #3

1.

```
etuning@cscd-linux01:~/netstorage/CSCD240$ nano printfDemo.c
etuning@cscd-linux01:~/netstorage/CSCD240$ gcc printfDemo.c -o printfDemo
printfDemo.c: In function 'main':
printfDemo.c:10:2: warning: incompatible implicit declaration of built-in function 'printf'
 printf("Size of short: %d\n", shortSize);
  ^
etuning@cscd-linux01:~/netstorage/CSCD240$ ./printfDemo
Size of short: 2
Size of int: 4
Size of char: 1
Size of long: 8
etuning@cscd-linux01:~/netstorage/CSCD240$
```

2. Use scanf

3.

```
etuning@cscd-linux01:~/netstorage/CSCD240$ ./getsFgetsDemo
Please enter a name: ethan
The name you entered is ethan
etuning@cscd-linux01:~/netstorage/CSCD240$ nano getsFgetsDemo.c
Use "fg" to return to nano.

[1]+  Stopped                  nano getsFgetsDemo.c
etuning@cscd-linux01:~/netstorage/CSCD240$ nano getsFgetsDemo.c
etuning@cscd-linux01:~/netstorage/CSCD240$ gcc getsFgetsDemo.c -o getsFgetsDemo
etuning@cscd-linux01:~/netstorage/CSCD240$ ./getsFgetsDemo
Please enter a name: ethan
The name you entered is ethan
etuning@cscd-linux01:~/netstorage/CSCD240$
```

gets() is unsafe to use because it does not know the size of the string so you could run into overflow errors. fgets() you have to list the size and the input explicitly making fgets() much better to use.

4. You will get the ASCII value of the “,” will be used as input and you will get incorrect results.

5. a = 4, b = 5, c = 8, size = 4