Lecture 6

```
void f1(int a, int b)
{
   int c = a;
   a = 5;
          
   b = 3;
}
void f2(int a, int b)
{
   int d = 8.2;
   char e = 'p';
   f1(a+3, b-7);
}
int main(int argc, char *argv[])
{
    int a = 12;
    int b = -9;
    printf("%d %d \n", a, b);
      f2(a, b);
    printf("%d %d \n", a, b);
}
```

← Draw the stack at this point

- C does not have call by reference ONLY call by value.

What happens when a function is called at run time?

- gcc marks return address (immediately after the call) and stores (called "push") it on the call stack
- gcc stores the address for the return value
- gcc copies the values to the arguments, stored on the call stack
- gcc allocates space for variables, stored on the call stack