Function Pointers

Pointers hold the address of some location in memory of an item. Functions are loaded in memory as well, so you can have a pointer which holds the beginning memory address of a function, i.e. a function pointer.

The syntax looks like the following:

<return type> (*fp_name)(parameter-list);

The following is a simple C program that is easy to understand. The same principle will be employed in 102 with objects in the raytracer program (Unit 6 Notes).

```
#include <stdio.h>

int adder(int a, int b) {
    return (a + b);
}

int main (void) {
    int sum;
    int (*ptrf) (int, int);  // declaring a function pointer
    // since the return type is an int, 
    // that is the type of pointer; 
    // the parameter types are required

    ptrf = adder;           // setting the function pointer to point
    // to the address location for adder()

    sum = (*ptrf)(3, 4);    // calling the function using the
    // function pointer;
    // the parentheses around the pointer
    // are required; the arguments need to
    // be the right types (both ints in this
    // case)

    printf("sum is: %d \n", sum);

    return 0;
}
```