Week 5: C Programming

Basic Data Types

- int
- float
- double
- char
- void
- no **bool** before C99

Pointers

- variables that store memory addresses

```
<variable_type> *<name>;

int *ptr;       // declare ptr as a pointer to int
int var = 77;   // define an int variable
ptr = &var;     // let ptr point to the variable var
```

Pointers to Functions

- aka **function pointers** or **functors**
- goal: write a sorting function
    - has to work for ascending and descending sorting order + other
- how?
    - write multiple functions
    - provide a flag as an argument to the function
    - polymorphism and virtual functions
    - use function pointers
- user can pass in a function to the sort function

```
// Declaration
double (*func_ptr)(double, double);
func_ptr = [&]pow;  // func_ptr points to pow()

// Usage
double result = (*func_ptr)(1.5, 2.0);
// same function call
result = func_ptr(1.5, 2.0);
```

```c
// qsort example

#include <stdio.h>
#include <stdlib.h>

int compare (const void * a, const void * b)
 {
    return ( *(int*)a - *(int*)b );
}

int main ()
{
    int values[] = { 40, 10, 100, 90, 20, 25 };
    qsort (values, 6, sizeof(int), compare);
    int n;
    for (n = 0; n < 6; n++)
        printf ("%d ",values[n]);
    return 0;
}
```

Dynamic Memory

- memory allocated at runtime
- allocated on the heap

```c
void *malloc (size_t size);
// allocates size bytes and returns a pointer to the allocated memory

void *realloc (void *ptr, size_t size);
// changes the size of the memory block pointed to by ptr to size bytes

void free (void *ptr);
// frees the block of memory pointed to by ptr
```

Reading/Writing Characters

```c
int getchar();
// returns next character from stdin

int putchar(int character);
// writes a character to the current position in stdout
```

Formatted I/O

```c
int fprintf(FILE *fp, const char *format, ...);
int scanf(FILE *fp, const char * format, ...);
// FILE *fp can be either
//   • a file pointer
//   • stdin, stdout, stderr
```

```
// the format string
int score = 120; char player[] = "Mary";
printf("%s has %d points.\n", player, score);
```