CPS393 Summary Part 2: C Programming
Printed 2011-10-23

Week 1

printf, scanf	Part of stdio.h %d for digit, %c char, %lf float, %s string, %p pointer **scanf returns the number of inputs it received**
putchar, getchar	outputs/gets single char
gcc	-c does not create executable, creates object files, not linked together -o creates an executable with the name that comes after -o parameter
make -f makefile Target: Dependency1 Dependency2 ->[tab] Rule	lab08Psep: lab08Pmain.o lab08PFunc.o gcc -o lab08Psep lab08Pmain.o lab08PFunc.o lab08Pmain.o: lab08Pmain.c lab08PFunc.h gcc -c lab08Pmain.c lab08PFunc.o: lab08PFunc.c lab08PFunc.h gcc -c lab08PFunc.c

Week 2

random number	<time.h>; srand(time(NULL)); int i = rand();
for, while, switch/case	
tolower isdigit isalpha	ctype.h
Static Functions and Variables	
gets	** Dangerous **
fgets	appends \n before the \0 null terminator if there is space (newline) \n has to be removed if you don't want the newline
strcpy, strcat, strlen	strlen returns length of string, everything up until \0 null terminator which is not counted
multidimensional arrays	
system("{command}");	system() executes any linux command

Week 3

arrays are always passed by ref	Arrays are already pointers, so when they are passed to other functions, they're addresses are passed.
Ways to make a string	char a[40]; strcpy(a,"hello"); OR char a[40] = "hello";

pointers	int *p; // this is a pointer to an int type
	int num = 3;
	p = # // assig the pointer by getting the address of a var using &
	printf("%d",*p); // * dereferences the pointer to get the value being pointed at

"multiple" pointers	char **mp, *p, ch; p=&ch; mp = &p; **mp = 'A'; // each * steps one deeper through the pointers

Week 4

int main(int argc, char *argv[])	argc receives the number of user arguments plus 1 for the program name argv is a vector which is an array of char arrays (strings). argv[0] is the char array which is the name of the progra argv[1] is the first user-inputted char array (string) argv[1][0] first char of the first user-inputted string
FILE (stdio.h) type	FILE *fp = fopen("myfile","r")
atoi strtof strtod	atoi convert string to int strtof convert string to float strtod convert string to double
fopen fclose	fopen("myfile","[r or w or a or r+ or w+ or a+]") returns a pointer to a FILE fclose(fp) closes the file associated with the pointer fp
fscanf fgetc fputc fgets fputs	fscanf(fp, "%d", i) fgetc(fp) returns char or EOF fputc(ch,fp) writes the ch, returns that ch if successful or EOF fgets(str,length,fp) fputs(str,fp)

Week 5

struct	Structure type (like a class, only primitive)
typedef	Creates custom types (useful for structures)

Week 6

calloc, malloc	dynamic array allocation; calloc -> multiple memory locations allocated; malloc -> one memory location: MyType *var = (MyType *)calloc(10, sizeof(MyType)) // pointer to the first one MyType *var2 = (MyType *)malloc(sizeof(MyType))
struct tm, time_t	Date/time functions, built-in