C Programing functions

# Classification

Include ctype.h

Int isalnum(int c);

Int isalpha(int c);

Int iscntrl(int c);

Int isdigit(int c);

­­­­­­int islower(int c);

int isprint(int c);

int ispunct(int c);

int isspace(int c);

int isupper(int c);

int isxdigit(int c);

int tolower(int c);

int toupper(int c);

# Math

Include math.h

Trig functions take double, return double.

* + e.g. cos(), sin(), tan()

double ceil(double x)

* + returns smallest integer not less than x

double floor(double x)

* + return largest integer not greater than x

Easiest to understand through an example:

If x is 5.3, floor of x is 5, ceil of x is 6

double fabs(double x)

* + absolute value

double pow(double x, double y)

* + power

double sqrt(double x)

* + square root

# Random

Include stdlib.h

Int rand(void);

void srand(unsigned int seed);

# System

Include stdlib.h

Int system(const char \*s)

Void exit(int status)

# Memory

Include stdlib.h

void \*memset(void \*p, int c, size\_t n)

* + Sets a memory block to the value of c.

void \*malloc(size\_t howMuch)

void \*calloc(size\_t howMany, size\_t howBig)

Initializes memory gotten to 0

free(\*)

void \*realloc(void \*ptr, size\_t size)

# String

Include string.h

int strncmp(const char \*s1, const char \*s2, size\_t n)

char \*strncpy(char \*s1, const char \*s2, size\_t n)

char \*strchr(const char \*s, int c)

char \*strrchr(const char \*s, int c)

* + strchr from the right

char \*strstr(const char \*s1, const char \*s2)

# Formatted strings

Printf %[flags][width][.precision][length]type

# Structs

struct Part

{

char partNumber[8];

char name[20];

double cost;

};

struct Part widget = { "123KDE9", "Grapplegrommet", 133.21 };

passing struct as parameters

The following are equivalent:

* + - (\*pPart).howMany = 9;
    - pPart->howMany = 9;

typedef struct

{

char partNumber[8];

char name[20];

double cost;

} Part;

* + In this example, Part refers to the struct.
  + In this case, you do not have to put the *struct* keyword when you declare variables of that data type.
  + e.g. Part myPart;

# File I/O

Include stdio.h

Include windows.h

FILE \*fp = NULL;

FILE \* fopen(char \*filename, char \*access);

if( (fp = fopen("c:\\test.txt", "r")) == NULL )

fprintf(fp, "%d %s\n", number, name)

if( feof(fp) != 0 ) // true if at end of file

if( ferror(fp) != 0 ) // true if there is an error

fclose(fp)

size\_t fread(void \*a\_ptr, size\_t element\_size, size\_t n, FILE \*fp);

WIN32\_FIND\_DATA fileData = {0};

HANDLE FindFirstFile(char \*where, WIN32\_FIND\_DATA \*pInfo)

If FindFirstFile() returns INVALID\_HANDLE\_VALUE, it didn’t find anything

FindNextFile() uses both the handle returned from FindFirstFile() and the WIN32\_FIND\_DATA variable

FindClose(), passing the handle returned from FindFirstFile()

if( (filedata.dwFileAttributes & FILE\_ATTRIBUTE\_DIRECTORY) != 0