CSE 333 Reference Sheet (Midterm)

C Library Header – `stdio.h`

```
FILE          // type of object containing info to control a stream

FILE*  fopen (const char* filename, const char* mode);
int    fclose (FILE* stream);
int    fprintf (FILE* stream, const char* format, ...);
char*  fgets (char* str, int num, FILE* stream);
size_t fread (void* ptr, size_t size, size_t count, FILE* stream);
size_t fwrite (const void* ptr, size_t size, size_t count, FILE* stream);
void   perror (const char* str);
int    ferror (FILE* stream);       // returns non-zero if error on stream
```

C Library Header – `stdlib.h`

```
EXIT_SUCCESS  // success termination code
EXIT_FAILURE  // failure termination code

void*  malloc (size_t size);
void*  realloc (void* ptr, size_t size);  // change size of mem block *ptr
void   free (void* ptr);                  // does nothing when ptr = NULL
void   exit (int status);                 // terminate calling process
```

C Library Header – `string.h`

```
size_t strlen (const char* str);           // # of chars, not including '\0'

char*  strcpy (char* dst, const char* src);          // copy chars
char*  strcat (char* dst, const char* src);          // append chars
int    strcmp (const char* str1, const char* str2);  // compare strings
```

- Versions that take a third parameter `size_t num`: `strncpy()`, `strncat()`, `strncmp()`

C Library Header – `math.h`

```
INFINITY      // Infinity
NAN           // Not-A-Number

float  abs (float x);                // absolute value
float  pow (float base, float exp);  // base raised to the power exp
float  sqrt (float x);               // square root
float  ceil (float x);               // round up (towards +∞)
float  floor (float x);              // round down (towards -∞)
```

- All of these functions are overloaded to work with `double`, too

POSIX Library Headers – `fcntl.h`, `unistd.h`, `dirent.h`

```
O_RDONLY      // read-only flag
O_WRONLY      // write-only flag
O_RDWR        // read-write flag
O_APPEND      // append (add to end) flag
DIR           // type representing a directory stream

int     open (char* pathname, int flags, ...);          // open a file
int     close (int fd);                                 // close a file
ssize_t read (int fd, void* buf, size_t count);         // read from file
ssize_t write (int fd, const void* buf, size_t count);  // write to file

DIR*    opendir (const char* dirname);                  // open a directory
int     closedir (DIR* dirp);                           // close a directory
struct dirent* readdir (DIR* dirp);                     // read a directory
```

Error Library – `errno.h`

```
errno         // # of the last error, usually checked against defined consts

EACCES        // permission denied
EBADF         // bad file/directory descriptor
EFAULT        // bad address supplied
EINTR         // interrupted function
EISDIR        // is a directory
ENOTDIR       // is not a directory
```

C++ Memory Allocation

```
new           // allocate space for type, return pointer
new[]         // allocate space for array of type, return pointer
delete        // deallocate space indicated by pointer
delete[]      // deallocate space of array indicated by pointer
```

Format Specifiers

Specifier	Type	Specifier	Type
d / i	signed decimal int	f	decimal float
u	unsigned decimal int	c / s	character / string
x	unsigned hex int	p	pointer address

Streams

<stdio.h>	POSIX	<iostream>
stdin	0	std::cin
stdout	1	std::cout
stderr	2	std::cerr

Makefile Automatic Variables

`$@  # target name`	`$^  # all sources`	`$<  # left-most source`