CSC 209H1 S 2015 Midterm Test
Duration — 50 minutes
Aids allowed: none

Student Number: |_|_|_|_|_|_|_|_|_|_|_|_|_|

Last Name: _____ **First Name:** _____

Lecture Section: L0101 **Instructor: Reid**

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm consists of 5 questions on 8 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments are not required, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code. Answers that contain both correct and incorrect or irrelevant statements will not get full marks.
If you use any space for rough work, indicate clearly what you want marked.

# 1:	_____/ 6
# 2:	_____/ 3
# 3:	_____/ 2
# 4:	_____/ 5
# 5:	_____/13
TOTAL:	_____/29

LEC 0101

Question 1. [6 MARKS]

Part (a) [1 MARK]

Write one line that you can type at the bash shell prompt to change to the parent directory of your home directory.

Part (b) [1 MARK]

Run the program `markall` found in the current directory so that it reads standard input from the file `classlist`, saves the standard output in a file `results`, and leaves standard error going to the console.

Part (c) [1 MARK]

Write a snippet of C code that prints "yes" if the bit at index 3 in the variable `perms` is set, and prints "no" otherwise.

```
unsigned short perms;
```

Part (d) [3 MARKS]

A user types the following on the command line:

```
red green blue | grey >& white
```

Fill in the table below for each of the processes in the command line. Note that there may be more columns than necessary.

Program name			
Argument(s)			
Standard input comes from			
Standard output goes to			
Standard error goes to			

Question 2. [3 MARKS]

Circle the correct answer, and briefly explain it.

TRUE FALSE A child process has the same address space as its parent.
Explain:

TRUE FALSE In a parent process, the wait() call blocks until at least one child signals the parent with its exit status.
Explain:

TRUE FALSE Global variables should never be declared in header files.
Explain:

Question 3. [2 MARKS]

Consider the following makefile:

```
simpletest : simpletest.o smalloc.o testhelpers.o
  gcc -Wall -g -o simpletest simpletest.o smalloc.o testhelpers.o

%.o : %.c smalloc.h
  gcc -Wall -g -c $<
```

The current working directory contains the following files

```
Makefile    simpletest.c    smalloc.c   smalloc.h   testhelpers.c
```

Part (a) [1 MARK] If we run `make`, which new files are added?

Part (b) [1 MARK] If we modify `smalloc.c` and then run `make` again, which files are modified?

Question 4. [5 MARKS]

Consider the following program:

```
int main() {

    if(fork() == 0) {
        if(fork() == 0) {
            fprintf(stderr, "B");
            exit(3);
        } else {
            fprintf(stderr, "B");
            exit(2);
        }

    } else {
        int status;
        fprintf(stderr, "C");
        if(wait(&status) != -1) {
            if(WIFEXITED(status)) {
                fprintf(stderr, "%d", WEXITSTATUS(status));
            }
        }
        fprintf(stderr, "D");
    }
    fprintf(stderr, "E");
    return 0;
}
```

Part (a) [1 MARK] How many processes including the original one are created? _____

Part (b) [3 MARKS]

Is more than one output ordering possible? If so, write all of the possible orders. If not, write the output, and explain why it must be the only order.

Part (c) [1 MARK]

Is it possible for a child process in this program to ever become an orphan? Briefly explain your answer.

Question 5. [13 MARKS]

Part (a) [8 MARKS] Complete the code below so that it is correct, and the `contacts` contains:

```
{"pizza pizza", "416-967-1111"}, {"ghostbusters", "555-3268"}
```

```
struct contact {
    char *name;
    char phone[16];
};
void set_contact(_________________________ c, char *new_name, char *new_phone) {









}
/* Add "code" to the beginning of the number in "num".  Assume that the caller
 * has already ensured that there is enough space in "num" to add "code". */
void add_area_code(char *code, char *num) {









}
int main() {
    struct contact *contacts = malloc(_____________________________________);

    // Set the the zero'th element of contacts


    set_contact(_____________________________, "pizza pizza", "416-967-1111");

    // Set the next element of contacts


    set_contact(_____________________________, "ghostbusters", "555-3268");

    // add area code to the phone number for "ghostbusters"


    add_area_code("416-", ____________________________);
}
```

Part (b) [5 MARKS]

Give the declaration for the variable `x` for each of the statements below, or if there is an error in the statements write "ERROR". Assume that the variables `contact` and `clist` are appropriately initialized and memory has been allocated.

```
struct contact *contact;
struct contact clist[5];
```

	Declaration for x
`x = &contact;`	
`x = contact[0].name[0]`	
`x = &(contact[1].phone)`	
`x = &(contact[1].phone[3])`	
`x = *contact`	

C function prototypes and structs:

```
int execlp(const char *file, char *argv0, ..., (char *)0)
int execvp(const char *file, char *argv[])
int fclose(FILE *stream)
char *fgets(char *s, int n, FILE *stream)
pid_t fork(void)
FILE *fopen(const char *file, const char *mode)
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
int fseek(FILE *stream, long offset, int whence);
      /* SEEK_SET, SEEK_CUR, or SEEK_END*/
size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
char *index(const char *s, int c)
void perror(const char *s);
unsigned int sleep(unsigned int seconds)
int sprintf(char *s, const char *format, ...)
int stat(const char *file_name, struct stat *buf)
char *strchr(const char *s, int c)
size_t strlen(const char *s)
char *strncat(char *dest, const char *src, size_t n)
int strncmp(const char *s1, const char *s2, size_t n)
char *strncpy(char *dest, const char *src, size_t n)
char *strrchr(const char *s, int c)
int wait(int *status)
int waitpid(int pid, int *stat, int options) /* options = 0 or WNOHANG*/
ssize_t write(int d, const void *buf, size_t nbytes);

WIFEXITED(status)      WEXITSTATUS(status)
WIFSIGNALED(status)    WTERMSIG(status)
WIFSTOPPED(status)     WSTOPSIG(status)
```

Useful structs

```
struct stat {
    dev_t st_dev; /* ID of device containing file */
    ino_t st_ino; /* inode number */
    mode_t st_mode; /* protection */
    nlink_t st_nlink; /* number of hard links */
    uid_t st_uid; /* user ID of owner */
    gid_t st_gid; /* group ID of owner */
    dev_t st_rdev; /* device ID (if special file) */
    off_t st_size; /* total size, in bytes */
    blksize_t st_blksize; /* blocksize for file system I/O */
    blkcnt_t st_blocks; /* number of 512B blocks allocated */
    time_t st_atime; /* time of last access */
    time_t st_mtime; /* time of last modification */
    time_t st_ctime; /* time of last status change */
};
```

Print your name in this box.

End of Examination