

# CS 2110 Quiz 5

Izaan Kamal

TOTAL POINTS

**54 / 100**

QUESTION 1

Types 20 pts

1.1 V 0 / 0

✓ + 0 pts Graded

✓ + 0 pts Correct: `$$\text{pointer to char}$$`

1.2 x[4] 2.5 / 5

✓ + 0 pts Graded

+ 5 pts Correct: `$$\text{pointer to char}$$`

No credit given if suggested `$$\text{x}$$` is an `$$\text{array of char}$$` instead of an `$$\text{array of pointer to char}$$`

Can also mention `$$\text{fifth element}$$` or `$$\text{fourth index}$$` `$$\text{of array x}$$`

✓ + 2.5 pts Unclear answer:

"`$$\text{pointer to char at ... position in array}$$`",

"`$$\text{...pointer to char in x}$$`", or equivalent

Unclear if interpreting `$$\text{x}$$` as `$$\text{array of char}$$` or `$$\text{array of pointer to char}$$`

- 1.5 pts Deduction: Indicated the element or index in the array but incorrect number

1.3 WWW 5 / 5

✓ + 0 pts Graded

✓ + 5 pts Correct: `$$\text{int}$$`

1.4 z[0] 5 / 5

✓ + 0 pts Graded

✓ + 5 pts Correct: `$$\text{pointer to pointer to float}$$`

1.5 \*\*y 0 / 5

✓ + 0 pts Graded

+ 5 pts Correct: `$$\text{pointer to float}$$`

QUESTION 2

Code Tracing 28 pts

2.1 \*ppc = &b; 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{pc = \&b}$$`

2.2 \*pb = 13 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{b = 13}$$`

2.3 \*\*ppb = c + 3; 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{b = 20}$$`

2.4 \*ppb = \*ppc; 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{pb = \&b}$$` or no change

2.5 pb = &c 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{pb = \&c}$$`

2.6 (\*\*ppc)++; 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{b = 21}$$`

2.7 \*\*ppb = b; 4 / 4

✓ + 0 pts Graded

✓ + 4 pts Correct: `$$\text{c = 21}$$`

QUESTION 3

### 3 Macros 7.5 / 10

+ 0 pts Graded

✓ + 5 pts Wrote a macro that compiles with the correct name and parameter:

\$\$\text{\#define PIE\\_AREA(radius) ...}\$\$

✓ + 2 pts Used outer parentheses:

\$\$\text{\#define PIE\\_AREA(radius) (...)}\$\$

✓ + 2 pts Encapsulated \$\$\text{radius}\$\$ with parentheses to ensure proper order of operations:

\$\$\text{... ((radius) \* (radius) ...)}\$\$

✓ + 1 pts The macro produces the correct output:

\$\$\text{((radius) \* (radius) \* (PI))}\$\$ or equivalent

✓ - 2.5 pts Used a semicolon

- 3 pts Used uppercase in \$\$\text{\#define}\$\$:

e.g. \$\$\text{\#DEFINE}\$\$

- 3 pts Used \$\$\text{=}\$\$:

e.g. \$\$\text{PIE\\_AREA(radius) = (...)}\$\$

- 3 pts Used different names for the parameter in the parameter list and in the expression

- 3 pts Used invalid symbol: \$\$\text{\pi}\$\$

- 3 pts Used invalid symbol: \$\$\text{PI()}\\$

- 2 pts Used incorrect operator: \$\$\text{\wedge}\$\$

- 2 pts Used invalid operator: \$\$\text{\*\*}\$\$

- 2 pts Used invalid operator: \$\$\text{^2}\$\$

- 2 pts Used parentheses to multiply

#### QUESTION 4

### Creating a Pumpkin Patch 42 pts

#### 4.1 Create a struct pumpkin 0 / 8

✓ + 0 pts Graded

+ 4 pts Created a \$\$\text{struct pumpkin}\$\$ that compiles:

\$\$\text{struct pumpkin \{}\$\$

\$\$\text{...}\$\$

\$\$\text{\};}\$\$

+ 2 pts Properly declared the following members:

\$\$\text{int seeds;}\$\$

\$\$\text{float weight;}\$\$

+ 2 pts Properly declared the character array:

\$\$\text{char name[10];}\$\$

- 2.5 pts Small syntax errors:

Missing semicolon, etc.

Improperly declaring struct, Improper declaration of int/float, random \* before the char[] declaration.

#### 4.2 typedef a pumpkin\_t 0 / 8

✓ + 0 pts Graded

+ 8 pts Correct:

\$\$\text{typedef struct pumpkin pumpkin\\_t;}\$\$

- 2.5 pts Missing semicolon

#### 4.3 Allocate a pumpkin\_patch 0 / 8

✓ + 0 pts Graded

+ 2 pts Allocated space with \$\$\text{malloc}\$\$

+ 2 pts Allocated the correct amount of space:

i.e. \$\$\text{sizeof(pumpkin\\_t) \* 20}\$\$

+ 2 pts Created a \$\$\text{pumpkin\\_t}\$\$

\*pumpkin\_patch and assigned it the address returned from \$\$\text{malloc}\$\$

+ 1 pts Performed a check for \$\$\text{malloc}\$\$ failure

+ 1 pts If \$\$\text{malloc}\$\$ failed: called

\$\$\text{exit(...)}\$\$, used \$\$\text{assert(...)}\$\$, called

\$\$\text{return 0;}\$\$, or something similar

✓ - 2.5 pts Small syntax error or runtime error

Syntax Error: pumpkin\_patch not defined, if statement wrong

#### 4.4 Initialize a pumpkin\_patch 4 / 10

✓ + 0 pts Graded

✓ + 4 pts Loops over the `$$\text{pumpkin\_patch}$$`

+ 1 pts No syntax errors

-- there should be twenty iterations

+ 3 pts Properly assigns zero to both  
`$$\text{seeds}$$` and `$$\text{weight}$$`

i.e. `$$\text{pumpkin\_patch}[i].seeds = 0;$$` or  
equivalent

+ 3 pts Properly assigns a zero-terminator to the  
first character of each `$$\text{name}$$` array

i.e. `$$\text{pumpkin\_patch}[i].name[0] = 0;$$` or  
equivalent

- 2.5 pts Small syntax errors:

Missing semicolon, etc.

- 6 pts Incorrect access operator for type:

`$$\text{.}$$` on `$$\text{struct*}$$`, `$$\text{->}$$` on  
`$$\text{struct}$$`, etc.

- 5 pts Unrecoverable error:

Segfaulting code, carryover error, incorrect type  
assignment, fundamental misunderstanding, etc.

#### 4.5 Name a pumpkin\_patch 2 / 8

✓ + 2 pts Graded (two points of free credit - yay!)

+ 0 pts Errors carried forward from previous parts  
(e.g. the type of the array) but this part is answered  
correctly when previous errors are followed

+ 2 pts Correctly modifies a struct on the array  
(dereference, arrow, dot, etc) -- don't worry about the  
index. No credit if making copy of array.

+ 3 pts OPTION 1: Attempts to assign individual  
characters (no string literal) -- give this credit even if  
they try but their assignment doesn't work for pointer  
etc. reasons

+ 2 pts OPTION 1: Attempts to assign null terminator  
-- give this credit even if they try but their assignment  
doesn't work for pointer etc. reasons

+ 5 pts OPTION 2: Used `strcpy` etc. to copy string  
(do NOT select together with any of the OPTION 1  
items)



This quiz is worth a total of 100 points.

In accordance with the Georgia Institute of Technology Honor Code, I have neither given nor received aid on this quiz.

Signature: Izaan Kamal

Please make sure all of your answers are contained within the answer boxes or the fill-in lines. You have been provided with scratch paper for your work. You will NOT be given credit for showing work. Having anything except the answer inside the boxes or above the fill-in lines might cause incorrect results. Write your name and answers legibly. You will not receive credit for illegible answers.

### Types

1. Consider the following C code segment:

```
char *v;
int **w, *ww, www;
char *x[15];
float *(*y)[];
float **z[10];
```

Please describe the evaluated type of the following expressions.

Note: (a) has been completed as an example.

(a) v pointer to char

(b) x[4] pointer to char at 5th index of array x

(c) ww integer

(d) z[0] pointer to pointer to float at 1st index of z

(e) \*\*y

5

5

5

5

### Code Tracing

2. For each line in the following table, show the updated value of the variable after the line is executed. You must have exactly one entry in each row. Use the & operator to denote the address of a variable.

Note: The first six lines have been filled for you!

Instructions	b	c	pb	pc	ppb	ppc
int b = 3;	3					
int c = 17;		17				
int *pb = &b;			&b			
int *pc = &c;				&c		
int **ppb = &pb;					&pb	
int **ppc = &pc;						&pc
*ppc = &b;				&b		
*pb = 13;	13					
**ppb = c + 3;	20					
*ppb = *ppc;			&b			
pb = &c;			&c			
(**ppc)++;	21					
**ppb = b;		21				

28



## Macros

3. Write a macro called `PIE_AREA` with parameter `radius` which calculates the surface area of a pie. 10

Recall that the area of a circle is  $\pi r^2$  where  $r$  is the radius.

Assume a macro `PI`, a symbolic name for 3.14159f, has been written on a previous line in the file.

```
#define PIE_AREA(radius) ((PI) * (radius) * (radius));
```

## Creating a Pumpkin Patch

4. Note: Assume `stdlib.h` and `assert.h` have been included.

Note: If there is insufficient space in the heap, terminate the program with an error!

- (a) Define a struct `pumpkin` with an `int` (`seeds`), a `float` (`weight`), and an array of ten `char` (`name`). 8

```
struct pumpkin { int seeds, float weight, char[10] name};
```

- (b) Make a new type name `pumpkin_t` which is an alias for `struct pumpkin`. 8

```
struct pumpkin_t = struct pumpkin;
```

- (c) Allocate space for an array of twenty `pumpkin_t` on the heap, and name a pointer to the first element of the array `pumpkin_patch`. 8

```
if (int *ip = malloc (pumpkin_patch * pumpkin_t[20]) == 0)
    return NULL;
```

- (d) Initialize each `pumpkin_t`: Set `seeds` and `weight` to zero. 10

Assign the first character in each name to be `'\0'` – you need not assign the other nine characters.

```
int i;
for (i = 0; i < 20; i++)
```

- (e) For the fifth `pumpkin_t` in the `pumpkin_patch`, using the allocated space from part (d), set the name to "Jack". 8