# Pre-Lecture 7

**Due** Sep 11 at 9am **Points** 11 **Questions** 6 **Available** until Sep 11 at 9am
**Time Limit** None **Allowed Attempts** 2

# Instructions

Take this quiz ***after you have watched the required videos and/or read the associated sections of the textbook***. See Lecture 7: Heterogeneous data structures.

You may attempt this quiz twice. Incorrect responses are marked after each attempt. Correct answers are revealed at the start of class for this lecture.

Carefully note the deadline for responses. Submissions are not accepted after the deadline, and there is no grace period.

This quiz was locked Sep 11 at 9am.

## Attempt History

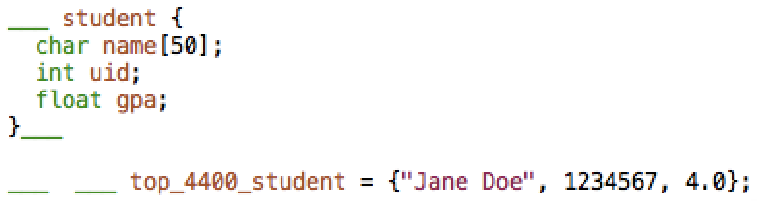
| | Attempt | Time | Score |
|---|---|---|---|
| **KEPT** | Attempt 2 | 2 minutes | 11 out of 11 |
| **LATEST** | Attempt 2 | 2 minutes | 11 out of 11 |
| | Attempt 1 | 9,066 minutes | 10.5 out of 11 |

Score for this attempt: **11** out of 11
Submitted Sep 10 at 11:26pm
This attempt took 2 minutes.

### Question 1

**1 / 1 pts**

Fill in the blanks to complete the structure declaration and use of the new type in a variable declaration.



```
___ student {
  char name[50];
  int uid;
  float gpa;
}___

___  ___ top_4400_student = {"Jane Doe", 1234567, 4.0};
```

(NOTES: This example does not use typedef. Fragments of ***exact*** C code are expected as answers. Blanks are ordered top to bottom, left to right.)

1st blank: struct

2nd blank: ;

3rd blank: struct

4th blank: student

**Answer 1:**

Correct!

struct

**Answer 2:**

Correct!

;

**Answer 3:**

Correct!

struct

**Answer 4:**

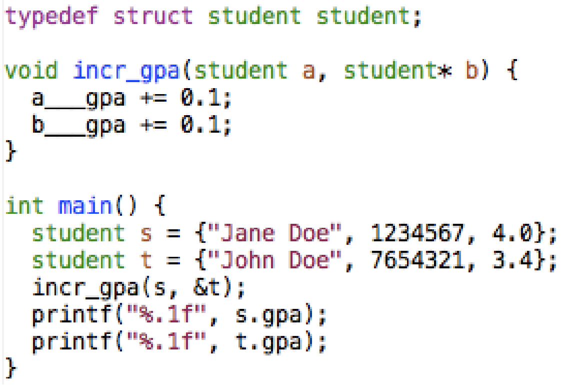
Correct!

student

## Question 2

**2 / 2 pts**

Consider again the structure declaration from Question 1. Suppose that we have added the following to make a program. Fill in the blanks with C operators to complete the function definition:



```
typedef struct student student;

void incr_gpa(student a, student* b) {
  a___gpa += 0.1;
  b___gpa += 0.1;
}

int main() {
  student s = {"Jane Doe", 1234567, 4.0};
  student t = {"John Doe", 7654321, 3.4};
  incr_gpa(s, &t);
  printf("%.1f", s.gpa);
  printf("%.1f", t.gpa);
}
```

1st blank: . 2nd blank: ->

Now, what is the output of main?

1st output: 4.0 2nd output: 3.5

(NOTES: %.1f simply restricts the printing to one digit on the right of the decimal point. You are permitted to check your answers by writing a C program; however, in order to prepare for exams, first attempt a solution without doing so.)

**Answer 1:**

Correct!

.

**Answer 2:**

Correct!

->

**Answer 3:**

Correct!

4.0

**Answer 4:**

Correct!

3.5

## Question 3

2 / 2 pts

Suppose that we have the following structure:

```
struct node {
  short x;
  int y;
  double z;
  struct node* next;
} n;
```

Also, suppose that n is stored at memory address 0. Fill the following blanks with ***decimal*** memory addresses:

n.x is at memory address 0

n.y is at memory address 4

n.z is at memory address 8

n.next is at memory address 16

**Answer 1:**

Correct! 0

**Answer 2:**

Correct! 4

**Answer 3:**

Correct! 8

**Answer 4:**

Correct! 16

## Question 4

**2 / 2 pts**

Suppose that we have the following structure:

```
struct my_struct {
  ___ a;
  ___ b;
  ___ c;
  ___ d;
} s;
```

Also, suppose that s.a is stored at memory address 0, s.b is stored at memory address 1, s.c is stored at memory address 16, and s.d is stored at memory address 20.

What could be the type of struct member a? char

What could be the type of struct member b? char[12]

What could be the type of struct member c? int

What could be the type of struct member d? int[2]

**Answer 1:**

Correct!

char

**Answer 2:**

Correct!

char[12]

**Answer 3:**

Correct!

int

**Answer 4:**

Correct!

int[2]

## Question 5

**1 / 1 pts**

Consider the following C code fragment:

```
typedef union {
  char c;
  int i;
  double d;
} my_union;

my_union arr[2];
```

Suppose that arr[0] is stored at memory address 0. At what address is arr[1]? (Give your answer in decimal.)

Correct!

8

orrect Answers

8

## Question 6

3 / 3 pts

Consider the following incomplete C code fragment:

```
union u1 {
  struct s1* h;
  struct s2* i;
  char j;
};

struct s1 {
  char a[3];
  union u1 b;
  int c;
};

struct s2 {
  struct s1* d;
  char e;
  int f[4];
  struct s2* g;
};

int proc(struct s1* x) {
  return x->___->___;
}
```

Suppose that the body of proc is translated to the following x86 code:

```
movq    8(%rdi), %rdi
movl    24(%rdi), %eax
```

Fill in the blanks to complete the C definition of proc.

1st blank: b.i

2nd blank: f[3]

(NOTE: ***Exact*** C code is expected.)

**Answer 1:**

Correct! b.i

**Answer 2:**

Correct! f[3]

Quiz Score: **11** out of 11