CSSE 120 – Introduction to Software Development in C

Quiz 3 from Videos:

Structs Point

Name: _____ **Section:** _____ **Grade:** _____

Answer these questions while viewing the assigned videos. ***Not sure of an answer?*** Ask your instructor to explain ***at the beginning of the next class session***. You can then fill in your answer, still for full credit. (But no fair doing that unless you attempted the question first.)

Videos for this quiz: www.rose-hulman.edu/class/csse/binaries/C-Videos/session4/

Video 1: Structs (7 minutes, 20 seconds)

Notes on the video:

- A ***structure*** in C is declared with the **`struct`** keyword. We call them ***structures*** or ***structs*** (two ways to say the same thing).
- ***Structures*** are a weak version of Python's ***classes*** – structures ***know*** things (i.e., a structure instance contains data) but they ***can't do*** things (i.e., there are no methods associated with them).
- The parts of a structure are called its ***members***, or its ***fields***, or its ***instance variables*** – different ways to say the same thing. (*C* tends to call them *members*, *Java* tends to call them *fields*, *Python* tends to call them *instance variables*, but they are all the same concept. I usually use the phrase *fields*.)
- The video and examples below use the types:
  - ***`double`*** -- just like a ***`float`***, but with more precision and larger possible magnitude.
  - ***`char`*** – character
  - array of ***`char`***'s – a ***`string`***

1. **`#define`** is often used to define named _____.

2. If we want to name a new type in C, we use the keyword_____.

3. The individual members of a structure instance are accessed using the _____ operator.

4. Suppose we had a structure definition like that shown to the right:

```
typedef struct {
    int age;
    float weight;
    char breed[30];
} Dog;
```

   a. What is the name of this structure type? _____

   b. How many fields does it have? _____

   c. Write a statement that declares a variable called ***`fido`*** of type ***`Dog`***.

d. Write statements that, assuming that a variable called ***fido*** has already been declared (per the previous subproblem), set the fields of ***fido*** so that it is a 3-year-old Brittany Spaniel that weighs 32.8 pounds.

```
typedef struct {
    int age;
    float weight;
    char breed[30];
} Dog;
```

5. **True** or **False** (circle one): The name of a structure type, defined with ***typedef***, can be used in variable declarations or formal parameter declarations, just like a built-in type (e.g., ***int***) is used.
6. Continuing to use the **Dog** structure from above, write a function called **heavier_dog** that has two parameters, both of which are **Dog** instances, and returns the Dog whose weight is greater. (If they have the same weight, the function returns the first-parameter Dog.)

Video 2: Point (a worked example of using structures) (6 minutes, 33 seconds)

Note: You can check out the project mentioned in this video as **Session28_C1_StructureExample**

7. **True** or **False** (circle one): The fields of a structure can themselves be structure types. (Hint: The answer is ***True***.)
8. Define a structure type called **LineSegment** that represents a line segment, given that a line segment has two **end points**. That is, your structure should have 2 (not 4) fields, each of which is a **Point**, where **Point** is as defined in the video.

If you have other questions about these videos /reading, ask them at your next class session!