Quiz Submissions - Quiz 4

Thi Da Tang (username: tang0181)

Attempt 1

Written: Mar 6, 2019 6:05 PM - Mar 6, 2019 6:17 PM

Submission View

Released: Mar 8, 2019 8:02 AM

Question 1 **0.5 / 1 point**

A "self-referencing structure" is instrumental in building which of the following types of collections?

- ➡ ✗ ☐ Doubly-linked lists
- ✓ ☐ Arrays
- ➡ ✓ ☐ Linked lists
- ➡ ✗ ☐ Trees

Question 2 **1 / 1 point**

Unions are fundamentally unnecessary in C because you can implement all polymorphic linked lists with a void * pointer instead of a union.

- ○ True
- ✓ ○ False

Question 3 **1 / 1 point**

You are creating a program that includes a catalog of books. You have a "struct Book" that includes all of the information about each book (e.g., author, price, # of pages, title, etc.).

You now need to implement your catalog as a linked list. Which is the better design approach?

- ○ Add a self-referencing "next" pointer (e.g., "`struct Book *next;`") to the existing "struct Book" definition
- ✓ ○ Create a new "struct Node" that includes just a pointer to a Book structure, and a self-referencing "next" pointer (e.g., "`struct Node *next`").

Question 4 **0 / 1 point**

How much memory will 'myStat' occupy?

```
union statistic {
    short age;
    short weight;
    short IQ;
    short height;
    char  gender;
} myStat;
```

Answer:

34 ✖ **(2)**

Question 5 **0 / 1 point**

What does assigning NULL to the variable 'pMyLinkedList' do?

```
Node *pMyLinkedList;
// some code
pMyLinkedList = NULL;
```

- automatically frees all elements previously assigned to the list
- ➡ indicates that our list has no elements in it
- ✖ shows that our list is undefined, and that a List object needs to be created before it can be subsequently populated with nodes

Question 6 **1 / 1 point**

It is possible to assign your own values to enums, e.g.,

```
enum colors { red=0xff000, blue=0x00ff00, green=0x0000ff };
```

- ✔ True
- False

Question 7 **1 / 1 point**

How many bytes are use to represent each enum value?

- 1 long
- 1 char
- ✔ It's up to the compiler, but generally it will be the smallest integer data type that can represent the largest value of the enum

Question 8 **1 / 1 point**

Which of the following is a syntactically-correct way to define a singly-linked list?

✔
```
typedef struct _node {
      Student *pStudent;
      struct _node *pNext;
} Node;
```

```
typedef struct _node {
      Student *pStudent;
      Node *pNext;
} Node;
```

```
typedef struct _node {
      Student *pStudent;
      struct Node *pNext;
} Node;
```

Question 9 **0 / 1 point**

We are implementing a function that appends a node to the end of a linked list. Our function has the signature

```
Node *appendNode( Node **ppFirstNode, Node *pNewNode );
```

This function is being called (from main) as follows

```
Node *pHead = NULL;
Node *pNode = appendNode( &pHead, createAndInitializeNode() );
```

Clearly, if our linked list is empty, then the node added by appendNode will become the first node in our list. Consequently, the appendNode function may have to update a pointer to the first node of the list (which is why we're passing in a reference to pHead).

So, within the implementation of appendNode, which is the correct way to update the head of the list?

```
*ppFirstNode = *pNewNode;
```

➡
```
*ppFirstNode = pNewNode;
```

✖
```
ppFirstNode = pNewNode;
```

```
ppFirstNode = *pNewNode;
```

```
&ppFirstNode = pNewNode;
```

```
ppFirstNode = &pNewNode;
```

Question 10 **0 / 1 point**

An enum is...

A collection of unique strings

✖ Automatically converted in to a collection of #define directives

A collection of unique integer values

Attempt Score:5.5 / 10

Overall Grade (highest attempt)**:**5.5 / 10

Done