Quiz 4 Version A

Due No due date **Points** 20 **Questions** 14
Available after Nov 9 at 3:30pm **Time Limit** None
Allowed Attempts Unlimited

Instructions

Quiz for Lecture 4: Linked Data Structures, Linked Lists, and const

18 points required to pass

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	less than 1 minute	0.5 out of 20 *

* Some questions not yet graded

Score for this attempt: **0.5** out of 20 *
Submitted Nov 28 at 2:07pm
This attempt took less than 1 minute.

Question 1 — 0.5 / 1 pts

Examine this code. Line numbers in main are specified in comments. In the check boxes below, check each line of code that WILL NOT COMPILE.

```
struct cl_node {   char data;   cl_node *next;   };

int main() {

    const double pi(3.141592654);        // 2
```

```
    const double e;                    // 3
    const cl_node n1{ 'a', nullptr };  // 4
    cl_node *p1 = &n1;                 // 5
    const cl_node *p2(&n1);            // 6
}
```

- [ ] Line 2 will NOT compile

orrect Answer

- [ ] Line 3 will NOT compile
- [ ] Line 4 will NOT compile

Correct!

- [x] Line 5 will NOT compile

p1 is a pointer-to cl_node, that we have initialized with the address of a cl_node const. This is not valid: you may not have a pointer-to non-const that points to a const. (If this were valid, it would be possible to change the const object indirectly via the pointer.) This statement will NOT compile.

- [ ] Line 6 will NOT compile

Jnanswered

Question 2

0 / 1 pts

Examine this code. Line numbers are specified in comments. In the check boxes below, check each line of code that WILL NOT COMPILE.

```
struct cl_node {   char data;   cl_node *next;   };

int main() {
```

```
    cl_node n1;                 // 2
    cl_node *p1{&n1};           // 3
    const cl_node *p2;          // 4
    p2 = &n1;                   // 5
    p1->data = 'B';             // 6
    *p1.next = 0;               // 7
    p2->data = 'x';             // 8
}
```

- [ ] Line 2 will NOT compile
- [ ] Line 3 will NOT compile
- [ ] Line 4 will NOT compile
- [ ] Line 5 will NOT compile
- [ ] Line 6 will NOT compile

orrect Answer

- [ ] Line 7 will NOT compile

orrect Answer

- [ ] Line 8 will NOT compile

Question 3

Not yet graded / 1 pts

Write the C++ code to define an **il_node** structure, in which the **data** member is an **int** and the **next** member is a *pointer-to* **il_node**:

Your Answer:

A correct answer:

```
struct il_node {
    int data;
    il_node *next;
};
```

Formatting details may vary.

Unanswered

Question 4

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

```
struct cl_node {   char data;   cl_node *next;   };
```

Which of these would be the most reasonable interface for a function that makes a singly linked list of **char** from a C-string?

- void mk_clist_from_Cstring(cl_node *p) { ... }
- void mk_clist_from_Cstring(const char *cstr) { ... }
- char *mk_clist_from_Cstring(const char *cstr) { ... }
- cl_node *mk_clist_from_Cstring(cl_node *p) { ... }

Correct Answer

- **cl_node *mk_clist_from_Cstring(const char *cstr) { ... }**
- cl_node **mk_clist_from_Cstring(const char *cstr) { ... }

- cl_node *mk_clist_from_Cstring(const char *cstr, cl_node **pp) { ... }

Unanswered

Question 5

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

struct cl_node { char data; cl_node *next; };

Which of these would be the most reasonable interface for a function that modifies the data in one or more of the linked list nodes, but does not modify the shape of the linked list?

Correct Answer

- **void clist_mod_data(cl_node *p) { ... }**
- void clist_mod_data(const cl_node *p) { ... }
- void clist_mod_data(cl_node **pp) { ... }

Unanswered

Question 6

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **double**:

struct dl_node { double data; dl_node *next; };

Which of these would be the most reasonable interface for a function that modifies the shape of the linked list (adding one or more nodes, deleting one or more nodes, changing the order of nodes, ...)?

- void dlist_change_shape(const dl_node *p) { ... }
- void dlist_change_shape(dl_node *p) { ... }

orrect Answer

- **void dlist_change_shape(dl_node **pp) { ... }**

Jnanswered

Question 7

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node { int data; il_node *next; };

Which of these would be the most reasonable interface for a function that accesses the data in one or more of the linked list nodes, but does not modify any data and does not modify the shape of the list?

orrect Answer

- **int ilist_access_data(const il_node *p) { ... }**
- int ilist_access_data(il_node *p) { ... }
- int ilist_access_data(il_node **pp) { ... }

Jnanswered

Question 8

0 / 1 pts

What is typically specified as the return type of a linked list interface function, if the value could be any non-negative integer?

- sizeof
- nullptr

orrect Answer

- **size_t**
- void *

Question 9

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node { int data; il_node *next; };

Write the body of code for this function, that returns the number of nodes in the list of integers:

size_t ilist_size(const il_node *p)

{

    // this is the part you have to write as your answer

}

Your Answer:

```
size_t sz(0);
for ( ; p; p = p->next)
    sz += 1;
return sz;
```

Question 10

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node { int data; il_node *next; };

Write the body of code for this function, that returns **true** if all the data values in the list are > 0, and **false** otherwise:

```
bool ilist_all_gt_0(const il_node *p)
{
    // this is the part you have to write as your answer
}
```

Your Answer:

```
for ( ; p; p = p->next)
    if (p->data <= 0)
        return false;
return true;
```

Question 11

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

struct cl_node { char data; cl_node *next; };

Write the body of code for this function, that returns the address of the first **cl_node** in a list of **char** that is a duplicate of the argument list of **char**.

```
cl_node *mk_clist_from_clist(const cl_node *p)
{
    // this is the part you have to write as your answer
}
```

Your Answer:

```
cl_node *p2(0);
cl_node **pp2(&p2);
for ( ; p; p = p->next) {
    *pp2 = new cl_node{ p->data, nullptr };
    pp2 = &(*pp2)->next;
}
return p2;
```

Question 12

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **double**:

struct dl_node { double data; dl_node *next; };

Write the body of code for this function, that inserts the argument **value** at the beginning of the list of **double**.

```
void dlist_insert_value(dl_node **pp, double value)
{
    // this is the part you have to write as your answer
}
```

Your Answer:

```
dl_node *p = new dl_node{ value, *pp };
*pp = p;
```

Question 13

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node { int data; il_node *next; };

Write the body of code for this function, that returns a pointer to the first **il_node** in a list of **int** values, where the values in the list are sequential integers in the range [**first**..**last**], that is, **first**, **first + 1**, **first + 2**, ..., **last**. (If **first** > **last**, the function should not create a list, and should return **nullptr**.)

```
il_node *mk_ilist_sequence(int first, int last)
{
    // this is the part you have to write as your answer
}
```

Your Answer:

```
if (first > last)
    return nullptr;
il_node *pfirst(nullptr);
il_node *plast(nullptr);

for (int i(first); i <= last; ++i) {
    il_node *p = new il_node{ i, nullptr };
    if (!pfirst) {    pfirst = plast = p;    }
    else {    plast = plast->next = p;    }
}
return pfirst;
```

Question 14

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **double**:

struct dl_node { double data; dl_node *next; };

Write the body of code for this function, that multiplies the current **data** value of each node by **factor**.

```
void dlist_multiply_by_factor(dl_node *p, double factor)
{
    // this is the part you have to write as your answer
}
```

Your Answer:

```
for ( ; p; p = p->next)
    p->data *= factor;
```

Quiz Score: **0.5** out of 20