

Computer Science 2530
Spring 2020
Practice Exam 4

The first two questions use the following structure type definition.

```
struct Feline
{
    int        size;
    const char* coat;
    Feline(int s, const char* c)
    {
        size = s;
        coat = c;
    }
};
```

1. [MC] Suppose that you have already created a variable called `cat` of type `Feline`. Which of the following statements will set the 'size' field of `cat` to hold 10?
 - (a) `s.Feline = 10;`
 - (b) `cat.s = 10;`
 - (c) `cat.size = 10;`
 - (d) `Feline.size = 10;`
 - (e) `size.cat = 10;`

2. [MC] Which of the following statements or sequences of statements will create a variable `p` of type `Feline*` and make `p` point to a new `Feline` structure whose 'size' field holds 8 and whose 'coat' field holds "tabby"?
 - (a) `new Feline* p(8, "tabby");`
 - (b) `Feline* p = new Feline*; p->size = 8; p->coat = "tabby";`
 - (c) `Feline* p = new Feline(8, "tabby");`
 - (d) `Feline* p(8, "tabby");`
 - (e) `Feline* p = new Feline*; size.p = 8; coat.p = "tabby";`

Types `ListCell` and `List` are as defined in class. Here are their definitions.

```
struct ListCell
{
    int      head;
    ListCell* tail;

    ListCell(int h, ListCell* t)
    {
        head = h;
        tail = t;
    }
};
typedef ListCell* List;
```

You can assume that constant `emptyList` and functions `isEmpty(L)`, `head(L)`, `tail(L)` and `cons(x, L)` have been defined as in class.

3. [MC] Which of the following will create variable L , of type `List`, and make it point to a new linked list holding 4 and 7, in that order? (Using our conceptual list notation, it must make L refer to list [4, 7].)
 - (a) `List L = new ListCell(4, new ListCell(7, NULL));`
 - (b) `List L = new ListCell(7, new ListCell(4, NULL));`
 - (c) `List L = new List(4, new List(7, NULL));`
 - (d) `List L = new ListCell(4, 7);`
 - (e) `List L = new ListCell(7, 5);`

4. Suppose that variables L and n have already been created. L has type `ListCell*` and points to a linked list of length three, and n has type `int`. Which of the following sets variable n to the second integer in list L ?
 - (a) `n = L->head->tail`
 - (b) `n = L->tail->tail;`
 - (c) `n = L->tail->head;`
 - (d) `n = L->2;`
 - (e) `n = L[1];`

5. Suppose that $\text{sum}(L)$ is intended to return the sum of the values in list L . For example, $\text{sum}([8, 2, 5]) = 8 + 2 + 5 = 15$ and $\text{sum}([9, 7]) = 9 + 7 = 16$. The sum of an empty list is 0.

(a) *Using the conceptual notation for lists discussed in class*, complete the following equations so that, taken together, they define $\text{sum}(L)$ for every list L . Use conceptual notation, not C++ notation, for this part. See the bottom of the last page for a brief summary of conceptual list notation.

$$\text{sum}([]) = \underline{\hspace{10em}}$$

$$\text{sum}(L) = \frac{\hspace{10em}}{\text{(when } L \neq [])}$$

(b) Following your equations from part (a) closely, write a C++ definition of $\text{sum}(L)$. It must not change any of the cells in list L . **Do not use any kind of loop for this definition.** A heading is given.

```
int sum(List L)
```

6. Suppose that function $\text{negatives}(L)$ is intended to be a nondestructive function that returns a list obtained from list L by replacing each value x by $-x$. For example,

- $\text{negatives}([3, 0, -9]) = [-3, 0, 9]$,
- $\text{negatives}([-3, 0, 9]) = [3, 0, -9]$,
- $\text{negatives}([4]) = [-4]$,
- $\text{negatives}([-4]) = [4]$,
- $\text{negatives}([]) = []$.

(a) *Using the conceptual notation for lists discussed in class*, complete the following equations so that, taken together, they define $\text{negatives}(L)$ for every list L . Use the examples above to help you work these out. **Do not guess. Do not use C++ notation.**

$$\text{negatives}([]) = \underline{\hspace{10em}}$$

$$\text{negatives}(L) = \underline{\hspace{10em}} \\ \text{(when } L \neq [])$$

(b) Demonstrate that your equations are correct by using them to compute, in order,

1. $\text{negatives}([]) =$

2. $\text{negatives}([5]) =$

3. $\text{negatives}([3, 5]) =$

Do not just write the answers without using your equations. If you discover that your equations are not correct then fix them and recompute each of the above.

- (c) Following your equations from part (a) closely, write a C++ definition of `negatives(L)`. **It must not change any of the cells in list L . Do not use any kind of loop for this definition.** A heading is given.

```
List negatives(List L)
```

Summary of conceptual list notation.

<code>[]</code>	is an empty list
<code>isEmpty([])</code>	is true
<code>head([2, 4, 6, 8])</code>	= 2
<code>tail([2, 4, 6, 8])</code>	= [4, 6, 8]
<code>2:[4, 6, 8]</code>	= [2, 4, 6, 8]