

There are 7 questions. Answer 4 or more of them correctly to get credit. (Yes, question 0 counts.)

Question 0

What is the fundamental rule of Cocoa memory management?

Question 1

name is an instance variable, an NSString. One of these setter methods for it contains a memory management error. Which code listing contains the error, and what is the error?

Code listing A:

```
- (void)setName:(NSString *)newName {
    [name autorelease];
    name = [newName retain];
}
```

Code listing B:

```
- (void)setName:(NSString *)newName {
    NSString *temp = [newName copy];
    [name release];
    name = temp;
}
```

Code listing C:

```
- (void)setName:(NSString *)newName {
    [name release];
    [newName retain];
    name = newName;
}
```

Question 2

What is the memory management error in the following code?

```
- (NSArray *)anArrayOfStuff {
    NSMutableArray *result = [NSMutableArray array];
    int i;

    for (i = 0; i < 100; i++) {
        Widget *obj = [[Widget alloc] initWithInt:i];
        [result addObject:obj];
    }

    return result;
}
```

Question 3

When is it appropriate to use `@property(assign)` to generate accessor methods for an instance variable?

Question 4

You are writing a class named `Widget`, which has the following interface:

```
@interface Widget : NSObject {
    NSString *name;
}
+ (id)widgetWithName:(NSString *)aName;
- (id)initWithName:(NSString *)aName;
@end
```

Write the code for `+ (id)widgetWithName:(NSString *)aName`, following memory management convention. (You should call `-initWithName:` as part of the implementation.)

Question 5

You are implementing a binary tree, based on a class called `Node`, with the following interface:

```
@interface Node : NSObject {
    Node *parent;
    Node *leftChild;
    Node *rightChild;
}
- (void)setParent:(Node *)aNode;
- (void)setLeftChild:(Node *)aNode;
- (void)setRightChild:(Node *)aNode;
@end
```

Here are the implementations of `-setParent:` and `-setLeftChild:`

```
- (void)setParent:(Node *)aNode {
    [parent autorelease];
    parent = [aNode retain];
}

- (void)setLeftChild:(Node *)aNode {
    [leftChild autorelease];
    leftChild = [aNode retain];
    [leftChild setParent:self];
}
```

What is wrong with this?

Question 6

What is wrong with this code? Hint: it is not a memory management *error*, but it is a problem caused by poor memory management practices.

```
- (void)printHugeMultiplicationTable {
    int i, j;

    for (i = 0; i < 100000; i++) {
        for (j = 0; j < 10000; j++) {
            int product = i * j;
            NSString *logString =
                [NSString stringWithFormat:@"%i, %i: %i", i, j, product];
            NSLog(@"%@", logString);
        }
    }
}
```