

# COMP 1006 – Introduction to Computer Science II

## Quiz 2, Winter 2015

This is a closed book test. No calculators are allowed. All questions should be answered directly on this sheet in space provided. You have 45 minutes to complete this test.

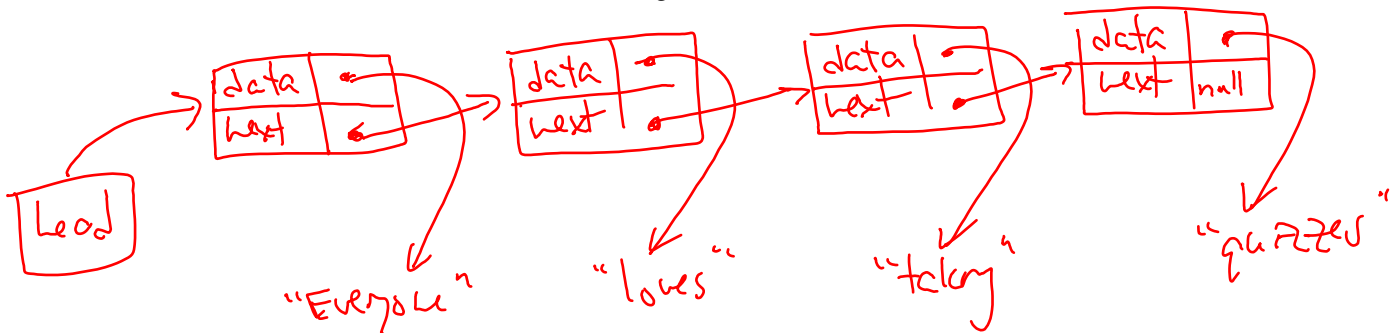
| Name      |    |   |    |   | TA Use Only |    |   |   |   |
|-----------|----|---|----|---|-------------|----|---|---|---|
| Student # | S  | O | L  | U | T           | I  | O | N | S |
| Tutorial  | B1 |   | B2 |   | Grade (/20) | TA |   |   |   |

1. Suppose we have a class named `Widget`. What is the result of saying `new Widget()` in Java compared to C++ [1 mark]? Where is the resulting object located in memory in Java compared to C++ [1 mark]? What happens when we are finished using the resulting object in Java compared to C++ [1 mark]?

Java: `new Widget()` returns a reference to the new object on the heap, which will be automatically deallocated when appropriate.

C++: `new Widget` returns a pointer to the new object on the heap, which must be manually deallocated using `delete`.

2. Suppose you have implemented a (singly) linked list in **Java**. The list stores the following Strings, starting at the head: "Everyone", "loves", "taking", "quizzes". Draw a diagram that shows how the linked list and its elements are represented in memory [3 marks]. Be sure to include all of the attributes described in class. Indicate references using arrows.



3. The following C++ code compiles without any errors, but there are at least **three** problems that occur at run-time. Identify and explain these problems [1 mark each].

```
int main() {
    int* a;
    *a = 5; ①

    int* b = new int;
    delete b;
    *b = 5; ②

    int* c = new int;
    *c = 5; ③

    return 0;
}
```

- ① no int has been allocated (statically or dynamically) for a to point to (i.e., a is uninitialized).
- ② The value \*b is being used after it has been deleted.
- ③ \*c has been allocated on the heap but not deleted.

4. What is the output of the following Java code [1 mark]? Why [2 marks]?

```
public class Widget {
    int x;
    static int s = 6;

    public Widget(int val) {
        this.x = val;
        System.out.println("1");
    }

    public Widget() {
        this(0);
        System.out.println("2");
    }

    public void change() {
        s++;
    }

    public static void main(String args[]) {
        ① Widget a = new Widget();
        ② Widget b = new Widget(3);
        ③ a.change();
        ④ System.out.println(b.s);
    }
}
```

Output:  
1  
2  
1  
7

- Reason:
- ① Constructor Widget() is called, which immediately calls Widget(0) ∴ "1" is printed out, then Widget(0) returns, and "2" is printed.
  - ② Widget(3) is called, which prints "1"
  - ③ The static variable s is changed from 6 to 7
  - ④ Widget b shares the value of s, so 7 is printed.

5. Create a **Java** class called `Bottle` that represents a bottle [1 mark]. It should have attributes to store the *contents* (a `String`), the *capacity* in milliliters (an `int`), and the *amount left* in milliliters (also an `int`) [1 mark]. Write two constructors: one that takes the contents, capacity, and amount left [1 mark]; and one that takes only the contents and capacity (and sets the amount left to be equal to the capacity) [1 mark].

Implement two overloaded methods called `drink`. Both should decrease the amount: one by taking a number of milliliters to decrease the amount left by [2 marks], and one by taking a percentage (a `float` between 0 and 1) of the **amount left** to decrease the amount left by [2 marks]. For example, if the amount left is 250, then calling `drink(50)` would cause the amount left to become 200 (since  $250-50=200$ ). Similarly, if the amount left is 100, then calling `drink(0.25f)` would cause the amount left to become 75 (since 25% of 100 is 25, and  $100-25=75$ ). There is no need to ensure that the amount left is non-negative. Note that Java will handle rounding any non-`int` answers for you.

```
public class Bottle {
    String contents;
    int capacity;
    int left;
    public Bottle(String c, int cap, int l) {
        this.contents = c;
        this.capacity = cap;
        this.left = l;
    }
    public Bottle(String c, int cap) {
        this(c, cap, cap);
    }
    public void drink(int amt) {
        this.left -= amt;
    }
    public void drink(float pct) {
        this.left *= 1 - pct;
    }
}
```