1. **What is Dependency Injection? Why is it important? Can you provide a concrete**

**example why an enterprise company should use DI?**

Answer: “A dependency is an [object](https://en.wikipedia.org/wiki/Object_(computer_science)) that can be used (a [service](https://en.wikipedia.org/wiki/Service_(systems_architecture))). An injection is the passing of a dependency to a dependent object (a [client](https://en.wikipedia.org/wiki/Client_(computing))) that would use it.” – (<https://en.wikipedia.org/wiki/Dependency_injection>)

**Why is it important?**

Answer:

* Highly configurable
* Loosely coupled
* Easier to test

**Can you provide a concrete example why an enterprise company should use DI?**

Answer: Better and easier test cases; runtime update of component configiurations.

1. **Compare unit tests with integration tests. Which is more appropriate? What is mocking and how can it be used in unit tests?**

Answer: Unit tests are targeted at the smallest software uinits (such as a class or method).  
Integration tests usually involve a slice across all layers of the system ; or simply tests multiple subsystems at once.

Mocking is creating Objects that are not able/ready to test currently, therefore we need to create a ‘Mocked’ up version for testing. (Note: in reality mocked objects usually break over time and lead to time spent maintaining unit tests).

1. **How do you find the first nonrepeated character in a string?**

*For example:*

*input: “aaabbccxyyz” → out: “x”*

*input: “abcabcabcxxz” → out: “z”*

**a. Write an algorithm in Java or Ruby to find the first nonrepeated**

**character?**

**b. What is the time complexity (Big O) of your solution? Explain why.**

**c. What is the maximum amount of disk space, in bytes, required for your**

**solution?**

Answer(A): Code also in ApplicationTest.java

Code taken from here…

http://javahungry.blogspot.com/2013/12/first-non-repeated-character-in-string-java-program-code-example.html

(b) O(N) – for n length and with 2 loops.

1. **Find if a given sum exists in an array of integers.**

**public** **static** **boolean** containsSum(**int**[] a, **int** sum){

HashMap<Integer, Boolean> map = **new** HashMap<Integer, Boolean>();

**for** (**int** i= 0; i< a.length; i++) {

map.put(sum - a[i], **true**);

}

**for** (**int** i = 0; i < a.length; i++) {

**if** (map.containsKey(a[i]) && map.get(a[i])) {

System.***out***.println("("+(sum-a[i])+","+a[i]+")");

**return** **true**;

}

}

**return** **false**;

Big 0 = O(N) for first and 2nd loops, answer: O(N)

(NB: referenced some from <http://stackoverflow.com/questions/576073>)