

# CICS 210 – Data Structures

---

*Duration: 2 hours / Without documents or calculator / Reply directly in this document*

Last name :	First Name :
-------------	--------------

Notes:

*No syntax error in the code (except if notified in the question)...*

*Only one answer for multi-choice questions*

## Question 1 (1 point)

Which naming convention is recommended in Java for variables?

- MyVariable
- myvariable
- my\_variable
- myVariable

## Question 2 (1 point)

Write a small Java program that calculates and displays the sum of the first 1000 positive integers using a “for” loop.

```
public class Main {  
    public static void main(String[] args) {  
        int sum = 0;  
        for (int i = 1; i <= 1000; i++) {  
            sum += i;  
        }  
        System.out.println("Sum first 1000 positive int: " + sum);  
    }  
}
```

### Question 3 (1 point)

What is the fundamental goal of Encapsulation in OOP?

Encapsulation aims to secure access to a class's data so that data declared as private can only be accessed and modified by methods defined within the same class.

### Question 4 (1 point)

Define an interface named `Vehicle` with a method `start()`. Then create a class `Car` that implements this interface and provides an implementation for `start()` which prints "START".

```
public interface Vehicle {  
    void start();  
}  
  
class Car implements Vehicle {  
    public void start() {  
        System.out.println("START");  
    }  
}
```

### Question 5 (1 point)

When examining the complexity of an algorithm, which two main factors are the focus of interest?

- The programming language used and the compiler speed
- Execution time and memory size
- Code quality and data type
- Number of connected peripherals and programmer quality

## Question 6 (1 point)

Are polynomial algorithms ( $O(n^k)$ ) considered effective? Why?

For small amounts of data, they are almost always fast enough  
but not large amounts of data.

## Question 7 (1 point)

What does the following code display, assuming a `TreeMap` named `arbreDepartements` is created from departments {1, "Ain"}, {2, "Aisne"}, {34, "Hérault"}, {38, "Isère"}?

```
int key = arbreDepartements.lowerKey(34);  
System.out.println(key);
```

2

## Question 8 (1 points)

Write the signature of a generic method named `createPair` that accepts two generic type arguments `A` and `B` and returns nothing (`void`).

```
public <A, B> void createPair(A first, B second)
```

## Question 9 (1 point)

What does “traversing (traversal) a binary tree” means?

It means to go through each node of the binary tree in a certain order to perform specific operations on it.

## Question 10 (1 point)

If you add successively 5, 3 and 7 into an empty BST with the “leaf” algorithm, which element becomes the root?

5

## Question 11 (1 point)

How is the balance factor of a node in an AVL tree defined?

- Total number of children.
- Height of the left subtree minus the height of the right subtree.
- Number of rotations performed.
- Maximum allowed height difference.

## Question 12 (1 point)

State one of the main conditions that a B-Tree of order  $m$  must satisfy regarding its non-leaf nodes.

All non-leaf nodes, except the root, must have at least  $m/2$  children.

## Question 13 (1 point)

Describe the two main steps required to insert a (key, priority) pair  $(K, P)$  into a Treap.

1. Insert the pair as a new leaf using the usual BST insert algorithm (based on key  $K$ ).
2. Rotate the node up using AVL rotations until the priority of its parent  
is greater than or equal to  $P$ .

### **Question 14 (1 point)**

State the fundamental property between nodes values in a Max-Heap.

**Every parent is greater-than both children**

### **Question 15 (1 point)**

What is the time complexity of the "sort-and-count" algorithm (based on Divide-and-Conquer and merge) for counting inversions?

- $O( n^2 )$
- $O( n \log(n) )$
- $O( n )$
- $O( \log(n) )$

### **Question 16 (1 point)**

What is the objective of the Closest Pair Problem?

**To find a pair of points in a plane with the smallest Euclidean distance between them.**

### **Question 17 (1 point)**

What is the principal difference in applying traversal methods (DFS/BFS) between directed and undirected graphs?

**The difference is that for a directed graph, the edge must be directed in the outgoing direction (meaning A must belong to  $\text{Deg+}(N)$  to visit vertex A adjacent to vertex N) and, for undirected graph, all edges are selected.**

## Question 18 (1 point)

In the case of an undirected graph, how is its Adjacency Matrix characterized?

- It is lower triangular
- It contains only edge costs
- It is symmetric with respect to the main diagonal
- It is always empty

## Question 19 (1 point)

What type of error can a Bloom filter produce, and what type of error can it never produce?

- It never produces false positives, but may produce false negatives
- It can produce false positives, but never false negatives
- It always produces errors
- It never produces errors

## Question 20 (1 point)

What does the following method return in Union-Find?

```
public boolean xxxx(int p, int q)
{ return find(p) == find(q); }
```

It returns true if nodes p and q are connected together.