



Systems Programming (English group)

Leganés, May 30, 2019

Duration: 40 min

Final exam ordinary call (test)

Score: 3 points out of 10 in the exam

There is only one correct option for each question. Each correct answer adds 0.15 points. Each incorrect answer subtracts 0.05 points. Unanswered questions do not add or subtract points.



- Mark the answer to each question with an “X” in the table below.
- If there is no “X” in a question or you mark more than one option, the question is considered unanswered.
- Complete **your personal information** before starting the exam.

Name:

Group:

Signature:

NIA:

--	--	--	--	--	--	--	--	--	--

	A	B	C	D		A	B	C	D
1					11				
2					12				
3					13				
4					14				
5					15				
6					16				
7					17				
8					18				
9					19				
10					20				



- 1.- Given the following code and variables `array = {1, 2, 3}` and `String b="hello"`, what is the value of the variables `array` and `b` after calling method `m(array, b)`?

```
public static void m(int[] array, String b){
    for(int i=0; i<array.length; i++){
        array[i] = array.length;
        b = "bye";
    }
}
```

- (a) *** `array = {3,3,3}` `b = "hello"`
- (b) `array = {1,2,3}` `b = "hello"`
- (c) `array = {3,3,3}` `b = "bye"`
- (d) `array = {1,2,3}` `b = "bye"`

- 2.- Given the following code, what will be the output after executing the `main` method?

```
public class Test{
    private static int a = 1;
    private static int b = 2;
    public static void main(String[] args){
        int a1 = a + b++;
        int a2 = ++a + 5;
        System.out.println(a1 + " " + a2);
    }
}
```

- (a) *** 3 7
- (b) 4 7
- (c) 3 6
- (d) 4 6

- 3.- Given the following code, indicate which method coverage is reached in the class `A` after executing the `main` method.

```
public class A{
    String a;
    public A(String a){setA(a);}
    public A() { this(null);}
    public setA(String a){this.a=a;}
}
public class B extends A{}
public class Test{
    public static void main(String[] args){
        B test = new B();
    }
}
```

- (a) *** 100%

- (b) 0 %
- (c) 33 %
- (d) 67 %

4.- Given the following code, indicate by which sentence would you need to replace *XXXX* to modify the value of the variable *a* so that it takes the value 5.

```
public class A{
    private static int a = 3;
    public void setA(int a){this.a =a;}
}
public class B extends A{
    public void modifyA(){
        XXXX
    }
}
```

- (a) ******* setA(5);
- (b) super.a = 5;
- (c) this.a = 5;
- (d) a = 5;

5.- Given the following code, indicate what modifiers need to replace *xxxx*, and *yyyy* to make the declarations of classes *A* and *B* correct.

```
public interface I1{void m1();}
public interface I2{void m2();}
public xxxx class A implements I1{ }
public yyyy class B extends A implements I2{
    public void m2(){...}
}
```

- (a) ******* xxx = abstract, yyy = abstract.
- (b) xxx = abstract, yyy no modifier
- (c) xxx no modifier, yyy = abstract
- (d) xxx no modifier, yyy no modifier

6.- Given the following code indicate which of the following statements about the constructors of the classes *A* and *B* is correct.

```
public class A{
    int a;
    public A(int a){this.a = a;}
}
public class B extends A{
    int b;
    public      B(int a, int b){
```

```

        super(a);
        this.b = b;
    }
}

```

- (a) *** The constructor of class A is invoked by the constructor of class B.
- (b) The constructor of class A is overloaded.
- (c) The constructor of class A is overridden.
- (d) Although not explicitly stated, class A has a default constructor that assigns the value 0 to the attribute a.

A implements I2

7.- Given interfaces I1, and I2 that inherits from the former and abstract class A and its non-abstract subclass B, indicate which of the following statements is *INCORRECT*.

- (a) *** A a = new A();
- (b) A a = new B();
- (c) I1 i1 = new B();
- (d) I2 i2 = new B();

8.- Indicate which of the following statements applies to white box testing.

- (a) *** Branch coverage is a type of white box test.
- (b) Equivalence classes define disjointed subsets of possible input data for white box tests.
- (c) White box tests can also be called input/output tests.
- (d) White box tests do not require knowledge of the internal structure of the code.

9.- Indicate what the following recursive method returns for a = 2.

```

public static int m(int a){
    int result = 0;
    if(a<=0){
        result = 3;
    }else{
        result = m(a-1) + m(a-2) - 2;
    }
    return result;
}

```

- (a) *** 5
- (b) 3
- (c) -2
- (d) 4

10.- Indicate which of the following statements about recursion is *correct*.

- (a) *** Tail recursion can be transformed in a loop.

- (b) Nested recursion can also be called cascade recursion.
- (c) Cascade recursion is more efficient than tail recursion.
- (d) Nested recursion is a case of linear recursion.

11.- If we have a non-empty binary tree represented as an array, knowing that, given any node in position i of the array, its left child is in position $2*i$ and its right child in position $2*i+1$, in which position of the array can we find the root of the tree?

- (a) *** 1
- (b) 0
- (c) $2*i - 1$
- (d) i

12.- The following binary tree, represented by the array {15,13,14,9,10,12,11,1,2,3,4,5,6,7,8} is:

- (a) *** Binary tree and max-heap.
- (b) Binary search tree.
- (c) Binary tree and min-heap.
- (d) Binary tree without being a heap or a search tree

13.- Given the class `LinkedListQueue<E>` which implements a queue without exception management, what is printed on screen after running this program?

```
public class Main{
    public static void main(String args []){
        LinkedListQueue<Integer> lq = new LinkedListQueue<Integer>();
        lq.enqueue(new Integer(10));
        lq.enqueue(new Integer(15));
        lq.enqueue(new Integer(20));
        Integer i = lq.dequeue();
        i = lq.front();
        i = lq.dequeue();
        i = lq.dequeue();
        i = lq.front();
        System.out.println(i);
    }
}
```

- (a) *** null
- (b) 20
- (c) 10
- (d) 15

14.- What data structure should be used to implement a non-priority queue?

- (a) *** Deque
- (b) Heap

- (c) Binary tree
- (d) Stack

15.- In an implementation of a doubly linked queue using dummy nodes, if we store N data in the N nodes with information of the queue, how many nodes does the queue have in total (including those that store information and those that do not)?

- (a) *** $N+2$
- (b) N
- (c) $N+1$
- (d) $N+3$

16.- A multinational has around 100,000 employees spread across the globe. The multinational has a proprietary software that stores the information of each of the employees, including first name, last name, date of birth, email account and username and password to access the computer. Knowing that the operation of login (via username and password) is the most frequent one and must be done as quickly as possible, what data structure should be implemented by the multinational proprietary software to store the information of its employees?

- (a) *** Binary search tree.
- (b) Linked list.
- (c) Heap as a priority queue
- (d) Stack

17.- Given the following interface, what does *top()* return if *isEmpty()* returns *true*?

```
public interface Stack<E> {  
    boolean isEmpty();  
    int size();  
    E top();  
    void push(E info);  
}
```

- (a) *** *null*
- (b) *new E()*
- (c) *throw new EmptyStackException()*
- (d) *throw new StackOverflowException()*

18.- We want to sort an array where $N = 1,000,000$ elements from lowest to highest. The array contains integers and is initially sorted from highest to lowest. What sorting algorithm should you choose looking for maximum efficiency?

- (a) *** HeapSort.
- (b) SelectionSort.
- (c) InsertionSort.

(d) QuickSort.

19.- Given the following sorted array of integers {6,12,15,16,20,25,30,38,40,50,60,61,67,80,85}, in how many iterations is number 61 located using binary search (iterative implementation)?

- (a) *** 2
- (b) 1
- (c) 3
- (d) 4

20.- Given the following code which prints on screen the contents of a linked list. If this method *print()* is called over a list of 10 items, what number of items does the list have after calling *print()*?

```
public void print(){
    if (first != null){
        Node<E> current = first;
        while (first!=null){
            System.out.println(first.getInfo());
            first = first.getnext();
        }
    }
}
```

- (a) *** 0
- (b) 10
- (c) -1
- (d) 1