



## Systems Programming Final exam (ordinary call)

Leganés, May 31, 2018

Duration: 40 minutes

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.- The reserved word `this` is used in Java to refer to:

- (a) \*\*\* An object.
- (b) A method.
- (c) An attribute.
- (d) A class.

2.- Considering the following code...

```
public class C{
    private static int a = 0;
    public static void main(String[] args) {
        System.out.println(a);
    }
}
```

- (a) \*\*\* The code is correct.
- (b) The code is incorrect because you cannot call a static attribute from a static method.
- (c) The code is incorrect because you have to create an object of the class *C* before you can use the attribute *a*.
- (d) The code is incorrect because the attribute *a* is private and cannot be accessed from the *main()* method.

3.- The modifier `final` applied on the parameter `a` in the following method...

```
public int m(final int a){...}
```

- (a) \*\*\* Indicates that the value taken by the parameter *a* when calling the method *m* cannot be modified inside that method.
- (b) Indicates that the method *m* cannot return *a*, so the *return a;* statement would give a compilation error.
- (c) Indicates that the method *m* cannot be overridden in the subclasses of the class where it is implemented.
- (d) Indicates that the memory space occupied by the parameter *a* will not be released when finishing the execution of method *m*.

4.- Given a class *A* which implements the interface *I*. Given *B* and *C* two classes which derive from *A*, and knowing that all classes have constructors which do not receive parameters, which of these statements is correct?

- (a) \*\*\* `I[] array = {new A(), new B(), new C()};`
- (b) `B[] array = {new A(), new B(), new C()};`
- (c) `C[] array = {new A(), new C(), new I()};`
- (d) `A[] array = {new B(), new C(), new I()};`

5.- The data structure `LinkedStack<Person>`, that is, a stack implemented with linked lists which stores information about persons can contain:

- (a) \*\*\* Objects of class `Person` and of any non-abstract class which derives from it.
- (b) Objects of class `Person` only.
- (c) Objects of class `Person` and of any interface implemented by the class `Person`.
- (d) Objects of classes `Person` and `Object`.

6.- Which of the following statements about constructors is correct?

- (a) \*\*\* All classes in Java have a default constructor which does not receive parameters, except in the case that other constructors have been explicitly programmed.
- (b) Constructors cannot be overloaded.
- (c) Constructors need to define a return type, even if it is *void*.
- (d) A constructor can call another constructor of the same class by using `this` and passing the name of the class in parentheses.

7.- In method overloading there are:

- (a) \*\*\* methods with the same name, but different number and/or type of parameters
- (b) methods with the same name, number and type of parameters
- (c) methods with different names, but the same number and type of parameters
- (d) methods with different names, number and type of parameters

8.- Given the following code and its test class, what line coverage is achieved?

```
public class C {
    public int m(int a, int b){
        if(a<0)
            return b;
        if(b<0)
            return a;
        return a+b;
    }
}

public class CTest {
    C c = new C();
    @Test
    public void test() {
        assertEquals(c.m(1, -1), 1);
        assertEquals(c.m(-1, 1), 1);
        assertEquals(c.m(1,1),2);
    }
}
```

- (a) \*\*\* 100 %.
- (b) 25 %.
- (c) 50 %.

(d) 75 %.

9.- What is the result of calling the following recursive method with values  $n = 10$ ,  $m = 2$ ?

```
public int m(int n, int m) {  
    if(n<m)  
        return n;  
    else  
        return 2*m(n-m, n+m);  
}
```

- (a) \*\*\* 16
- (b) 4
- (c) 6
- (d) 10

10.- What is the type of recursion of the following method, which is implemented in a recursively defined binary tree, where each root node has references to left (`getLeft()`) and right (`getRight()`) sub-trees, which may or may not be empty (`isEmpty()`)

```
public int m() {  
    if(isEmpty())  
        return 0;  
    else  
        return root.getLeft().m() + root.getRight().m() + 1;  
}
```

- (a) \*\*\* Cascade non-linear recursion.
- (b) Nested non-linear recursion.
- (c) Tail linear recursion.
- (d) Non-tail linear recursion.

11.- Given the following code, what does method `m()` return?

```
public class MyBasicLinkedList<E> {  
    protected Node<E> first;  
    private int mR(Node<E> list) {  
        if (list == null) return 0;  
        else return 1 + mR(list.getNext());  
    }  
    public int m() {  
        int a = mR(first);  
        return a;  
    }  
}
```

- (a) \*\*\* The size of the linked list

- (b) 0
- (c) 1
- (d) The sum of the values of all the elements in the linked list

12.- After running the following method, passing a stack (*stack*) as parameter:

```
void process(Stack stack) {
    Stack aux = new Stack();
    while (!stack.isEmpty()) {
        aux.push(stack.pop());
    }

    while (!aux.isEmpty()) {
        stack.push(aux.pop());
    }
}
```

- (a) \*\*\* The *stack* will contain the same items as before, in the same order.
  - (b) The *stack* will contain the same items as before, in the reverse order.
  - (c) The changes made to the *stack* in the method will not be visible because in Java the parameters of the methods are passed by value.
  - (d) The *stack* will be empty.
- 13.- To program a priority queue in Java it is necessary that the class of the objects used as key in the queue implement:
- (a) \*\*\* Interface **Comparable**
  - (b) Interface **Queue<E>**
  - (c) Interface **Deque<E>**
  - (d) Interface **Equals**
- 14.- What double-ended queue (*deque*) methods could be used to implement *push* and *pop* methods of a stack?
- (a) \*\*\* *insertFirst* for *push* and *removeFirst* for *pop*.
  - (b) *insertLast* for *push* and *removeFirst* for *pop*.
  - (c) *insertLast* for *pop* and *removeLast* for *push*.
  - (d) *insertFirst* for *push* and *removeLast* for *pop*.
- 15.- La interfaz **BTree<E>** interface contains a method **toString()** which returns the content of the tree in inorder, what is the method **toString()** exactly doing?
- (a) \*\*\* It returns a String with the content of the left child recursively, then the content of the root node, then the content of the right child recursively
  - (b) It returns a String with the content of the right child recursively, then the content of the root node, then the content of the left child recursively

- (c) It returns a String with the content of the children nodes recursively and then the content of the root node
  - (d) It returns a String with the content of the root node and then the content of the children recursively
- 16.- Given the following binary search tree [50,40,60,30,45,55,70], what is the sequence obtained as result of an in-order traversal?
- (a) \*\*\* 30, 40, 45, 50, 55, 60, 70
  - (b) 50, 40, 30, 45, 60, 55, 70
  - (c) 30, 45, 40, 55, 70, 60, 50
  - (d) 30, 45, 55, 70, 40, 60, 50
- 17.- Given the following binary tree [1, 2, 4, 3, 5, 6, 9, 8, 7], which statement is correct?
- (a) \*\*\* It is a min-heap
  - (b) It is not a heap, even though it meets the condition of minimum key in the root.
  - (c) It is a max-heap
  - (d) It is not a heap, even though it meets the condition of maximum key in the root.
- 18.- Given the following array [1, 2, 3, 4, 5, 6, 7, 8, 9]. How many iterations does the BinarySearch algorithm need to find number 5?
- (a) \*\*\* It finds it in the first iteration.
  - (b) It finds it in the second iteration.
  - (c) It finds it in the third iteration.
  - (d) It finds it in the fourth iteration.
- 19.- Given the SelectionSort sorting algorithm which looks for the minimum of the unsorted part and swaps it with the first unsorted element, and the following array, which is being sorted from lowest to highest by the algorithm, currently after its third iteration [3, 8, 15, 25, 20, 45, 35], how will the array look after the fourth iteration?:
- (a) \*\*\* 3, 8, 15, 20, 25, 45, 35
  - (b) 3, 8, 15, 25, 20, 45, 35
  - (c) 3, 8, 15, 20, 25, 35, 45
  - (d) 3, 8, 20, 15, 25, 45, 35
- 20.- Given an unsorted array of a thousand elements, which of the following algorithms is, on average, the most efficient one to sort the array?
- (a) \*\*\* Merge Sort
  - (b) Bubble Sort
  - (c) Selection Sort
  - (d) Insertion Sort