



Systems Programming (English group)

Leganés, March 9, 2018

First midterm exam (test)

Duration: 20 min

Score: 3 points out of 10 in the exam

There is only one correct option for each question. Each correct answer adds 0.3 points. Each incorrect answer subtracts 0.1 points. Unanswered questions do not add or subtract points.

Mark:		Cancel:		Do not use:			
<ul style="list-style-type: none">■ 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					6				
2					7				
3					8				
4					9				
5					10				



1.- Given the following code, what is printed on screen?

```
public class P {
    private int[] array = {};
    public static void main(String[] args){
        P p = new P();
        for (int i = 0; i <= p.array.length; i++){
            System.out.println(p.array[i]);
        }
    }
}
```

- (a) *** An *ArrayIndexOutOfBoundsException* is thrown
- (b) 0
- (c) *null*
- (d) A *NullPointerException* is thrown

2.- We are given the following code in two separate classes, what does it print on screen?

```
public class P {
    protected static int p = 0;
    public P(){
        p++;
    }
}
public class H extends P{
    public static void main(String[] args){
        P a = new P();
        H b = new H();
        System.out.println(p);
    }
}
```

- (a) *** 2
- (b) 0
- (c) 1
- (d) There is a compilation error and the program cannot be executed

3.- Complete the part of the code marked with ellipsis (...)

```
public class P {
    public static int divide(int a, int b) {
        return a / b;
    }
    public static void main(String[] args){
        try{
            System.out.println(divide(4,0));
        } catch (...){
            System.err.println("Error");
        }
    }
}
```

```

        }
    }
}

```

- (a) ******* *ArithmeticException arithmeticException*
- (b) *throw ArithmeticException*
- (c) *throws ArithmeticException*
- (d) *throw new ArithmeticException()*

4.- Attributes defined as *final*...

- (a) ******* Cannot change their values during the program
- (b) Can only be private
- (c) Must be initialized in the constructor of the class where they are defined
- (d) Cannot be overridden in subclasses

5.- Given classes *C1*, *C2* and *C3*, and interfaces *I1* and *I2*, which of these statements is correct?

- (a) ******* public abstract class C1 extends C2 implements I1
- (b) public abstract class C1 implements I1, I2 extends C2
- (c) public abstract class C1 extends C2, C3
- (d) public abstract interface I1 extends I2

6.- Given class *A*, which has a constructor which does not receive any parameters, and a constructor which receives one parameter, we want to call the first constructor from the second constructor, which sentence shall we use?

- (a) ******* *this()*
- (b) *super()*
- (c) *this.A*
- (d) *A()*

7.- Given abstract class *A*, its non-abstract child class *B*, and non-abstract class *C* which extends from *B*, all of them implementing a constructor which does not receive any parameters, indicate the statement which causes a runtime error:

- (a) ******* *C c = (C) new B();*
- (b) *A a = new B();*
- (c) *A a = new A();*
- (d) *B b = new C();*

8.- Considering interface *I*, which is implemented by classes *A* and *B*, with *A* an abstract class and *B* a non-abstract class we can say that:

- (a) *** Methods defined in I must be implemented in B , but not necessarily in A
- (b) A is a superclass of B
- (c) B must implement all the abstract methods in A or be declared as abstract
- (d) It is possible to create objects of A and B and store them in an array of type I

9.- Given the following class and its test class, which is the branch coverage achieved?

```
public class A {
    public boolean a(int a) {
        if (a <= 0) {
            return false;
        } else if (a > 100) {
            return false;
        } else
            return true;
    }
}

public class ATest {
    A a;
    @Before
    public void setUp() throws Exception {
        a = new A();
    }

    @Test
    public void testA() {
        assertTrue(a.a(10));
    }
}
```

- (a) *** 50 %
- (b) 25 %
- (c) 33 %
- (d) 100 %

10.- We are given a method which calculates the division of two integer values received as parameters *int division (int a, int b)*, and for which we want to carry out black-box testing. Select the pairs of values which allow testing all the equivalence classes for this method.

- (a) *** (4, 3) and (5, 0)
- (b) (0, 4) and (1, -1)
- (c) (1, -1) and (1, 1)
- (d) (-2, 0) and (2, 0)