

**Programming – Final Exam
January 2010
Leganés**



Universidad
Carlos III de Madrid

Ingeniería Informática

**READ CAREFULLY THESE INSTRUCTIONS BEFORE
STARTING THE EXAM:**

- Fill in all the pages with a pen (personal data and answers)
- Do not use a pencil or a red pen
- Do not forget your NIA and your actual group
- The duration of this exam is 2 hours and 30 minutes
- Books and notes are allowed. Laptops, mobile phones, or any other electronic device are not allowed.
- Use only these sheets for the answers of the questions. Use the back of the pages if needed. Additional sheets will not be considered in this first part.
- For the problems, use the provided sheets.

DO NOT CONTINUE WITH THE NEXT PAGE until indicated

Surname	Name	
Signature	NIA	Group

PART I: QUESTIONS

Question 1 (1 Point).- **Explain** if the following statements are true or not:

- 1.1. (0,25 points) We can execute a previously compiled class (`Class1`) with the Java SDK command: `javac Class1`.

False. `javac` is the JDK program to compile a class (i.e., to convert it into bytecode in a `.class` file). The program to run a class is `java`.

- 1.2. (0,25 points) Java does not perform automatically any conversion between basic data types.

False. Java automatically casts compatible datatypes. In general, a *larger* datatype is compatible with a *smaller* datatype. For instance, integers (`int`) are automatically casted to floating point values (`double`, `float`).

- 1.3. (0,25 points) A variable declared inside a method can be used from inside any other different method, including the main method.

False. The visibility of a variable is the block of code in which it has been defined, in this case, the method. It is possible to define variables global to all the methods of a class, but in this case, they must be declared out of a method.

- 1.4. (0,25 points) When calling a method, the value of the expression in the call is copied into the parameter of the method.

True. All parameters in Java are passed by value, which means that a copy of the value in the call to the method is copied into the parameter of the method.

Question 2 (1 Point).- Explain the output (i.e., what is printed on the screen) resulting from the execution of the `main` method of `Question2` class.

```
public class Question2 {
    public static void main(String[] args) {

        int [] badNumbers = new int[] {4, 8, 15, 16, 23, 42};

        for(int n = 1; n <= 50; n++) {
            boolean isBad = false;

            int i;
            for(i = 0; i < badNumbers.length; i++) {
                if(n == badNumbers[i]) {
                    isBad = true;
                    break;
                }
            }
            if(isBad) {
                System.out.println(
                    "Reached position "+ i + ", " + n+
                    " IS a bad number!");
            } else {
                System.out.println(
                    "Reached position "+ i + ", " + n+
                    " IS NOT a bad number!");
            }
        }
    }
}
```

This program checks if the numbers from 1 to 50 are elements of the array *badNumbers*. The output of the program is:

For each $i \in \{1, 2, \dots, 50\}$:

$i \notin \text{badNumbers} \rightarrow$ Reached position 6, i IS NOT a bad number

$i \in \text{badNumbers} \rightarrow$ Reached position k , i IS a bad number, being k the index of i in the *badNumbers* array (starting at 0)

```
Reached position 6, 1 IS NOT a bad number!
Reached position 6, 2 IS NOT a bad number!
Reached position 6, 3 IS NOT a bad number!
Reached position 0, 4 IS a bad number!
Reached position 6, 5 IS NOT a bad number!
Reached position 6, 6 IS NOT a bad number!
Reached position 6, 7 IS NOT a bad number!
Reached position 1, 8 IS a bad number!
Reached position 6, 9 IS NOT a bad number!
...
Reached position 6, 14 IS NOT a bad number!
Reached position 2, 15 IS a bad number!
Reached position 3, 16 IS a bad number!
Reached position 6, 17 IS NOT a bad number!
...
Reached position 6, 22 IS NOT a bad number!
Reached position 4, 23 IS a bad number!
Reached position 6, 24 IS NOT a bad number!
...
Reached position 6, 41 IS NOT a bad number!
Reached position 5, 42 IS a bad number!
Reached position 6, 41 IS NOT a bad number!
...
Reached position 6, 50 IS NOT a bad number!
```

Question 3 (1 Point).- Find and explain the 4 compilation errors included in the following Java code. **Explain** how they can be solved.

```
public class Question3 {  
  
    public static void method1 (String b) {  
        b = "Hello!";  
        return b;  
    }  
  
    public static int method2 (String a, double b) {  
        a = "Bye!";  
        b = 34;  
        int c = b+5;  
    }  
  
    public static void method3 () {  
        int a = 35;  
        double b = a;  
        short c;  
        System.out.println(a);  
    }  
  
    public static void main(String[] args) {  
        String h = "Bonjour!";  
        method1(h);  
        method2(h, 231.1);  
        method3(35);  
    }  
}
```

A void method cannot return any value. Solution is remove sentence "return b;".

A return instruction must be included. Solution is to add a return sentence; e..g, "return c".

method3 does not accept any parameter. The call to the method in the main should be changed to method3();

c is an integer, but b (and therefore b + 5) is a float. An explicit casting is required: int c = (int) (b+5);

Question 4 (1 Point).- We know that the code below prints on the screen:

hollo & byo

Explain which one of the following methods is actually called in the main method.

```
public class Question4 {

    public static void main(String[] args) {
        String a = "hello";
        String b= "bye";
        String c = method(a, b);

        System.out.println(c);
    }

    /** Creates a new string that is equal to 's' but
     *  replaces all the occurrences of the character
     *  'a' by the character 'b'
     *
     *  @param s String
     *  @param a Character to replace
     *  @param b Replacing character
     *  @return String with a replaced by b
     */
    public static String replace(String s, char a, char b) {
        return s.replace(a, b);
    }
}
```

a) `public static void method(String a, String b){`
 String c = a + " & " + b;
 c = `replace`(c, 'e', 'o');
 System.out.println (c);
 }

b) `public static String method(String a, String b {`
 String x = b + " & " + a;
 x = `replace`(x, 'e', 'o');
 return x;
 }

c) `public static void method(String a, String b){`
 c = a + " and " + b;
 c = `replace`(c, 'e', 'o');
 }

d) `public static String method(String a){`
 return "hollo & byo";
 }

None of the previous answers is correct.

a) is not correct, because the returning type is void. It would rise a compilation error, since the call `method(a,b)` in the main expects a String result.

b) is not correct, because the order of a and b is reversed (This answer was considered as almost valid in the exam evaluation)

c) is not correct, because c has not been declared. It would raise a compilation error.

d) is not correct, because it only accepts a parameter, whereas the call to `method` in the main uses two parameters. It would raise a compilation error.

PART 2: PROBLEMS

Problem 1 (1 Point).- Create a method named `countEqual` that, given two 2-dimension arrays of characters as parameters, returns the number of elements which are equal and are stored in the same position of both arrays. Assuming that the number of rows and columns of both arrays is the same, the method must work with any array size.

Example:

Given the arrays $\begin{bmatrix} 'a' & 'b' & 'c' \\ 'd' & 'e' & 'f' \end{bmatrix}$ and $\begin{bmatrix} 'z' & 'b' & 'y' \\ 'd' & 'x' & 'w' \end{bmatrix}$, `countEqual` must return the value 1.

Problem 2 (4 Points).

a) (1 Point) Create a method named `largest1D` that, given a 1-dimension `String` array as parameter, returns the largest string stored in it. (If there is more than one *largest* string, the method must return the first of them).

Example:

Given this definition:

```
String [] s = new String [] {"lion", "dog", "mouse", "rat"};
```

The call to the method `largest1D(s)` must return the string "mouse".

b) (1.5 Point) Create a class named `Problem2` with a `main` method that uses the `largest1D` method. The number of elements of the `String` array passed as parameter to `largest1D` must be an argument to the `main`. The actual values of the `String` array must be introduced by the keyboard.

c) (1.5 Point) Based on the `largest1D` method, create a method named `largest2D` that, given a 2-dimension `String` array as parameter, finds the largest string of each row, and returns a `String` array with these large strings.

NOTE: Remember that each row of a 2-dimension matrix is a 1-dimension array.

Example:

Given this definition:

```
String [][] s = new String [][] {
    {"lion", "dog", "mouse", "rat"},
    {"leg", "bones", "arm", "chest"},
    {"poppy", "daisy", "geranium", "rose"}
};
```

The call to the method `largest2D(s)` must return the 1-dimension array of `String` {"mouse", "bones", "geranium"}.

Problem 3 (1 Point).- Write a method to order an array of `double` values in decreasing order (i.e., the largest value must be in the first position of the solution array).

Programming – Grado en Ingeniería Informática

Authors

Of the English version:

Juan Gómez Romero

Based on the work by:

Ángel García Olaya

Manuel Pereira González

Silvia de Castro García

Gustavo Fernández-Baillo Cañas

Daniel Pérez Pinillos

Javier Ortiz Laguna

Álvaro Torralba Arias de Reyna



Universidad
Carlos III de Madrid