EXAM 2 PROGRAMMING **COMPUTING ENGINEERING** **Leganés**	 Universidad Carlos III de Madrid

Surname(s)	***Name***	
Signature	***NIA***	***Group***

READ THESE INSTRUCTIONS CAREFULLY BEFORE YOU BEGIN WITH THE EXAM:

- **Fill in all the sheets with a blue or black pen, both the personal information and the answers.**
- **Make sure that you have provided your NIA and the actual group you belong to.**
- **You have one hour to complete the exam.**
- **During the exam the only materials allowed on the table is the exam paper and the pen.**
- **Use only the exam sheets for your answers. Any additional papers will not be collected.**

Part 1: Questions

Question 1 (1 mark).- Indicate whether the following statement is correct, and briefly explain why.

"The following is a valid constructor declaration."

```
class Inverse {
      public void Inverse () {}
}
```

False, void is a type and a constructor should have no return value.

Question 2 (1 mark).- Indicate whether the following statement is correct and briefly explain why.

"A variable declared as protected is visible outside the class"

True, a variable declared as protected is visible to all the classes of the same package.

Question 3 (1 mark).- Why the value of n2 is 3 and not 4? How can we correct the code?

```
public class Main {
	public static void main (String [] args) {
		Counter n1 = new Counter ();
		Counter n2 = new Counter ();
		n2.print ();
	}
}

class Counter {
	private int c = 2;
	public Counter () {
		c += 1;
	}

	public void print () {
		System.out.println ("Result: " + c);
	}
}
```

Its 3 and not 4, because every object is unique, and so are their attributes. If we defined c to be static then we would have the desired result.

Question 4 (1 mark).- Find the errors in the following code and suggest a way of resolving them.

```
public class Main {

	public static void main (String [] args) {
		Question n1 = new Question (1,'c');
	}
}

class Question {
	int a = 0;
	static char c;

	public Question (int a, char c) {
		this.a = a;
		Question.c = c;
	}

	public void Question (char c, int a) {
		this.a = a;
		this.c = c;
	}
}
```

There are no errors in the above code.

Question 5 (1 mark).- In which ways do a generic method and a constructor defer?

A constructor should never have a return value. Moreover, a constructor should always inherit the name of the class. Finally, a constructor is the very first method executed upon object creation.

Question 6 (1 mark).- What does the term “class instantiation” describe?

The above term describes the creation of a new object.

Question 7 (1 mark).- How are arguments passed to methods? What is the value of x after executing the following code?

```
public class Main {

	public static void main (String [] args) {
		Flip f = new Flip ();
		boolean x = true;
		f.flip (x);
	}
}

class Flip {
	public void flip (boolean x) {
		x = !x;
	}
}
```

Arguments are passed to methods by value. That means that a copy of the value of a variable is made when that variable is passed as an argument to a method call. As a result the value of x in the example above is true.

Question 8 (1 mark).- What is the difference between the methods foo and bar of the following code?

```
public class Main {

	public static void main (String [] args) {
		Question q = new Question ();
	}
}

class Question {
	public static void foo () {}
	public void bar () {}
}
```

foo is defined as static hence it can be invoked directly through the class. bar on the other hand can be invoked always through some object.

PART 2: PROBLEMS

Problem 1 (2 marks).- Write a function which given two sets of integers returns a set of all the elements that the previous two sets have in common (i.e. their intersection). For example

```
Intersection in = new Intersection ();
int [] a = {1,2,3,4,5};
int [] b = {2,10,4};
int [] d = {20};
int [] foo = in.intersection (a,b);
int [] bar = in.intersection (b,d);
```

The value of *foo* should be {2,4}, while the value of *bar* should an empty array {}.

```
public int [] intersection (int [] s1, int [] s2) {
  int [] result = new int [Math.min (s1.length, s2.length)];
  int k = 0;

  for (int i = 0; i < s1.length; i++) {
      for (int j = 0; j < s2.length; j++ ) {
        if (s1[i] == s2[j]) {
            result[k] = s1[i];
            k++;
            break;
        }
      }
  }

  return result;
}
```