TYPE C252

1. (1.5p) a) Present the mechanism that allows us to use an instance of a derived class instead of an instance of its base class.

b) Does the next source code exhibit problems? Properly explain your answer.

```
class A {
        protected String value;

        void A(String value) {
                this.value = value;
        }
}
```

```
class B extends A{
        protected String value;

        B(String value) {
                super();
        }
}
```

c) Instantiate a Java array (after the element type insert []) of String that may contain 12 elements. Add only 2 elements in the array and iterate through the elements and print for each element its length (the length of each element is returned by the length() method belonging to class String).

2. (1p) Considering the source code given below, answer the following questions and properly explain your answer. a) What is going to be printed after the execution of Line 1? b) What is going to be printed after the execution of Line 2?

```
class MyClass {
    private int atr;
    private static int sAtr;

    public void set(int a, int s){
        this.atr = a;
        this.sAtr = s;
    }

    public int getAtr() {
        return atr;
    }

    public int getSAtr() {
        return sAtr;
    }
```

```
public static void myMethod(MyClass c1, MyClass c2) {
  c1 = c2;
  c1.set(18, 25);

  c2 = new MyClass();
  c2.set(6, 12); }

  public static void main(String[] args) {
     MyClass a = new MyClass();
     a.set(1, 2);
     MyClass b = new MyClass();
     b.set(11, 22);

     myMethod(a,b);
     System.out.println(a.getAtr() + " " + a.getSAtr());  //L1
     System.out.println(b.getAtr() + " " + b.getSAtr());  //L2
  }}
```

3. (1p) Considering the source code given below, answer the following questions and properly explain your answer. a) What is going to be printed after the execution of Line 1? b) What is going to be printed after the execution of Line 2?

```
import java.util.*;
class MyLecture {
  private String lecture;

  public MyLecture(String lecture) {
    this.lecture = lecture; }

  public boolean equals(Object l) {
    return (l instanceof MyLecture &&
        ((MyLecture)l).lecture.equals(lecture));
}
```

```
public static void main(String[] args) {
  List<MyLecture> s1=new ArrayList<MyLecture>();
  s1.add(new MyLecture("OOP"));
  s1.add(new MyLecture("OOP-Lab"));
  s1.add(new MyLecture("OOP"));
  System.out.println(s1); //Line1

  HashSet<MyLecture> s2;
  s2 = new HashSet<MyLecture>();
  s2.add(new MyLecture("OOP"));
  s2.add(new MyLecture("OOP-Lab"));
  s2.add(new MyLecture("OOP"));
  System.out.println(s2); //Line2
}}
```

4. (1p) Given the next class,

```
class Person {
        private String firstName, lastName;
        private int age;
}
```

Modify the Person class to include a constructor that ensures a Person object can only be instantiated if the age is between 0 and 120 (inclusive). No additional changes to the class are required.

5. (1p) How is cloning supported in Java? What is the difference between shallow and deep cloning?

6. (0.5p) We have inside a method a reference to an object whose type is String, as it follows:
`String s = "MyString";`
Present the changes we have to perform in order to modify the content of the existing String, from `"MyString"` to `"MyNEWString"`.

7. (0.5p) Define the meaning of encapsulation within the object-oriented paradigm.

8. (1p) Considering the source code given below, are the next statements true of false? a) Lines 1 and 2 generate compilation errors because interfaces cannot contain attributes; (b) Line 3 generates a compilation error because a class cannot implement more than one interface; (c) Line 4 does not generate a compilation error. Explain in detail your choices.

```
interface I1 { String S = "Interface1"; } //Line 1

interface I2 { String S = "Interface2"; } //Line 2

class C implements I1, I2 { //Line 3
        public String toString() { return S; }} //Line 4
```

9. (0.5p) Given the next class,
```
abstract class Person {
        private String firstName, lastName;
        // … constructor, toString,
}
```
do you think it is possible to instantiate the next array:
`Person[] persons = new Person[10];`
How many objects are stored inside the array?

10. (1p) Classes E1 si E2 are exceptions. What condition should be fulfilled in order to be able to compile the source code given below? Attention - the source code given below will not be altered!

```
        class SomeClass {
                public void foo(boolean b1, boolean b2) {
                        if (b1 && b2) throw new E1();
                        else throw new E2();
                }
        }
```