TYPE L201

1. (1p) Given the next code located in the same file, answer the next questions and properly explain your answers: a) What is going to be printed after the execution of Line1? b) What is going to be printed after the execution of Line2? c) Taking into account that the type of the object associated to a String key is Integer, do you think it is possible to add a primitive value (an int value) as performed in Line3?

```
import java.util.HashMap;
class P1_1a {
 public static void main(String[] args) {
   HashMap<String, Integer> myMap;
   myMap = new HashMap<String, Integer>();
   myMap.put("OOP-Jan20", new Integer(8));
   myMap.put("OOP-Feb20", new Integer(9));

   System.out.println(myMap); //Line1
   myMap.put("OOP-Feb20", new Integer(10));
   System.out.println(myMap); //Line2
 }
}
```

```
class P1_1b {
 public static void main(String[] args) {
  HashMap<String, Integer> myMap;
  myMap = new HashMap<String, Integer>();

  myMap.put("OOP-Feb20", 10); //Line3
 }
}
```

2. (0.5p) Given the next code, what is going to be printed out after the execution of Line1? Properly explain your answer.

```
import java.util.*;
class P1_2 {
 public static void service(Set<String> mySet) {
  mySet.add("Feb20");
  mySet = new HashSet<String>();
  mySet.add("Jan20");
  mySet.add("Feb20");
}
```

```
 public static void main(String[] args) {
  HashSet<String> mySet;
  mySet = new HashSet<String>();
  P1_2.service(mySet);
  System.out.println(mySet); //Line1
 }
}
```

3. (1p) Answer the following questions and motivate the given answers.
a) Is it wise to overload the well-known equals method from the Object class instead of overriding it?
b) What is the clone of an object?
c) What is the main difference between shallow and deep cloning?
d) How many methods named main can we define inside a class?

4. (0.5p) Without changing the provided lines of code, add inside the given class the needed service/services that allows/allow the clients of this class to get as many instances of this class they need.

```
class P1_4 {
        private P1_4() {}
        protected void service() {}}
```

5. (1p) Given the next code defined into a single file, answer the next questions and properly explain your answers: a) Does L1 generate a compilation error? If it does, get rid of it without changing the behaviour of the class! b) What is wrong inside P1_5c?

```
class P1_5a {
 private int attributeA;

 public P1_5a createObject(P1_5a object) {
   P1_5a newObject = new P1_5a();
   newObject.attributeA = object.attributeA; //L1
   return newObject; }
}

class P1_5b extends P1_5a {
 private int attributeB;

 public P1_5b(int attributeB) {
   this.attributeB = attributeB; }
}
```

```
class P1_5c extends P1_5b {
 private int attributeC;

 public P1_5c createObject(P1_5c object) {
  P1_5c newObject = new P1_5c();
  newObject.attributeB = object.attributeB;
  newObject.attributeC = object.attributeC;
  return newObject;
 }
}
```

6. (0.75p) What is going to be printed out after the execution of the next main method? Explain your answer.

```
class P6 {
    private int value;

    public P6() { value = getValue(); }

    public int getValue() { return 10; }

    public String toString() {
        return "Object value " + value;
    }
}
```

```
class P6D extends P6 {
    public int getValue() { return 20; }

    public static void main(String[] args) {
        P6 object = new P6D();
        System.out.println(object);
    }
}
```

7. (0.5p) Change the implementation of the constructor of the given class (the code inside its body placed between { . . . }) in order to make sure that the class is instantiated if and only if a positive number denoting its number of pages and a non null author are provided as parameters.

```
class Book7 {
    private int numberOfPages;
    private String author;

    public Book7(int numberOfPages, String author) {
        this.numberOfPages = numberOfPages;
        this.author = author;
    }}
```

8. (0.75p) Write the necessarily code that tests if the two defined collections store the same objects.

```
class P8 {
    public static void main(String[] args) {
        ArrayList<String> c1 = new ArrayList<String>(12);
        ArrayList<String> c2 = new ArrayList<String>(8);
    }
}
```

9. (1p) Given the next code, do the commented lines generate compilation errors? Explain your answers!

```
class P9 {
    public static void main(String[] args) {
        P9 p = new P9();
        System.out.println(p.toString()); //Line1
        p.main(null); //Line2
        P9.main(null); //Line3
        System.out.println(this.toString()); //Line4
    }
}
```

10. (1p) Change the marked lines (Line1, Line2 and Line3) in order to make them compilable. The behaviour of the given code should be preserved. Explain the performed changes!

```
class MyException extends Exception {}
class P10 {
    public void myService10() throws MyException {} //Line1
    public static void main(String[] args) {
        P10 object = new P10();
        try {
            object.myService10();
        }
        catch(Exception e) {} //Line2
        catch(MyException e) {} //Line3
    }
}
```

11. (0.5) How many access specifiers can we set to a class? Describe the meaning of each existing type.

12. (0.5) What is an immutable class? Present the actions that should be done in order to define an immutable class.