Name:	Number:

Object Oriented Programming 2015/16

Final Exam
June 28, 2016

Directions (read carefully):

- CLEARLY print your name and ID on every page.
- The exam contains 8 pages divided into 4 parts. Make sure you have a complete exam.
- The point value of each problem is indicated next to the problem and below.
- You have two hours.
- It is wise to skim all the problems before beginning, to best plan your time.
- This is a closed book exam. No notes of any kind are allowed. Do all work in the space provided, using the backs of sheets if necessary.
- **Over the table it should ONLY be this exam, a pen and an ID.**
- Turn off the mobile phone. The use of a mobile phone annuls the exam.

Part	Problem	Description	Page	Marks
I	1 a) b)	UML	2	1.5
II	2 a) b) c) d) e)	Java development	3	3.0
	3 a) b) c) d) e)	Java multiple choice	6	2.5
	4	Java miscellaneous	7	1.0
III	5	XML	8	1.0
IV	6	Swing	8	1.0
Total				**10.0**

Version D

Name:	Number:

Part I -- UML (1.5 marks)

1 – Consider a veterinary hospital that treats cats and dogs. The hospital maintains a list of owners and vets; both must contain a name. The owners also store an address; the vets store the license. An owner may have several pets; for each pet the system should store the pet birth date, the pet name, the pet ID and the pet owner. Whenever a pet arrives to the hospital it is seen by a vet; in this appointment the pet may receive a treatment. A treatment should identify the disease diagnosed, the medicine to apply and the date of treatment.

a) **[1.0 mark]** Define the UML class diagram for the presented problem.

Name:	Number:

b) [0.5 marks] Define an UML object diagram considering a owner of a dog and a cat and a vet (with no treatment). Set all needed attributes/associations with some dummy values.

Part II -- Java (6.5 marks)

2 – Consider an interface `Shape` with a no-arg method `area` that returns a double. In addition, consider the interface `Relatable` with a predicate `isBiggerThan` receiving another `Relatable` and returning a boolean. The method of `Relatable` should compare any two `Relatable` objects (strings vs strings, numbers vs numbers, shapes vs shapes, etc); whenever objects are not relatable it should throw an `ObjectsNotRelatable` exception. When comparing shapes, `isBiggerThan` should compare them by their area. For instance, a rectangle is bigger than a rectangle/triangle if its area is greater that the rectangle/triangle's area. Provide the implementation of classes `Rectangle` and `Triangle` that implement both interfaces. Note: you might use more classes than those directly asked.

a) **[0.25 marks]** Provide the `Shape` interface.

b) **[0.25 marks]** Provide the `Relatable` interface.

c) **[1 mark]** Provide the `Rectangle` class, with all needed fields and methods, and three constructors: a no-arg constructor, a copy constructor and a constructor that receives all fields needed to be initialized.

d) **[1 mark]** Provide the `Triangle` class, with all needed fields and methods, and three constructors as before.

e) **[0.5 marks]** Provide a `Main` class with a `main` method where a `Rectangle` and a `Triangle` are built and compared by `Relatable`. Print the result to the terminal.

Name: Number:

```
public class ObjectsNotRelatable extends Exception {}


public interface Shape {
     double area();
}


public interface Relatable {
     boolean isBiggerThan(Relatable r) throws ObjectsNotRelatable;
}


public abstract class RelatableShape implements Shape, Relatable {

     @Override
     public boolean isBiggerThan(Relatable r) throws ObjectsNotRelatable {
       if (r instanceof Shape){
              Shape s = (Shape) r;
              if (area()>s.area()) return true;
              else return false;
       } else
              throw new ObjectsNotRelatable();
     }
}


public class Rectangle extends RelatableShape {

     private int width, height;

     public Rectangle(){
       width = height = 1;
     }

     public Rectangle(Rectangle r){
       width = r.width;
       height = r.height;
     }

     public Rectangle(int w, int h){
       width = w;
       height = h;
     }

     @Override
     public double area() {
       return width*height;
     }
}
```

Name: Number:

```
public class Triangle extends RelatableShape {

     private int height;
     private int base;

     public Triangle(){
       height = 1;
       base = 1;
     }

     public Triangle(Triangle t){
       height = t.height;
       base = t.base;
     }

     public Triangle(int h, int b){
       height = h;
       base = b;
     }

     @Override
     public double area() {
       return height*base/2;
     }
}


public class Main {

     public static void main(String[] args){
       Triangle t = new Triangle(5,5);
       Rectangle r = new Rectangle(10,10);

       System.out.println("Triangle area: "+t.area());
       System.out.println("Reactangle area: "+r.area());

       try {
              System.out.println(r.isBiggerThan(r));
              System.out.println(r.isBiggerThan(t));
              System.out.println(t.isBiggerThan(r));
       } catch(ObjectsNotRelatable ex){
              System.out.println("Objects not relatable!");
       }
     }
}
```

Name:	Number:

3 – Fill the answers of multiple-choice questions in the following table (use only capital letters). If you want to correct your answer scratch out and write the correct answer. Each correct question is marked 0.5 points. A question not answered is marked 0 points, whereas a wrong answer discounts 0.2 points. If you think NONE of the options are correct, write NONE.

Question	a)	b)	c)	d)	e)
Answer	B	B	C	A	A

a) **[0.5 marks]** Which of the following statements is true?

A. Methods declared in interfaces are implicitly private.
B. Fields declared in interfaces are implicitly public, static and final.
C. Fields are not allowed to be declared in interfaces.
D. The keyword *implements* indicates that an interface inherits from another.
E. An interface can only extend one interface.

b) **[0.5 marks]** Which of the following are legal array declarations?
(i) `int i[5][];` (ii) `int i[][];` (iii) `int[][] i;` (iv) `int i[5][5];`

A. Only (ii)
B. Both (ii) and (iii)
C. Both (i) and (iv)
D. Three of them: (i), (ii) and (iv)
E. All of them: (i), (ii), (iii) and (iv)

c) **[0.5 marks]** Which of the following statements related to the garbage collector is correct?

A. Garbage collector ensures that the program never runs out of memory.
B. It is possible for a program to undelete an object already claimed by the garbage collector.
C. The `System.gc()` method does not guarantee that the garbage collector runs at that moment.
D. The `finalize` method in `Object` needs to be implemented if one wants to free an object.
E. It is not possible for a program to free memory in Java.

d) **[0.5 marks]** How many objects are eligible for garbage collection after executing line 7?

A. 0
B. 1
C. 2
D. 3
E. It crashes in run-time

```
public class Tester {
  public static void main(String[] args) {
    Integer x = new Integer(3000);
    Integer y = new Integer(4000);
    Integer z = new Integer(5000);
    Object a = x;
    x = y;
    y = z;
    z = null; //line 7
  }
}
```

Name: Number:

e) **[0.5 marks]** What is the result of compiling and running the following code?

A. 10
B. 11
C. 20
D. 21
E. None of the previous

```
public static void main(String[] args) {
   int x = 0, y = 0;
   if ((y == ++x) && (x < ++y)) {}
   System.out.println(x + "" + y);
}
```

4 – [1.0 marks] Explain, using an UML diagram, the inheritance relationship between parameterized types in java. That is, how `Object`, `Number` and `Integer` are related? How `Set<?>`, `Set<Object>`, `Set<Number>`, `Set<Integer>`, `Set<? extends Integer>` and `Set<? extends Number>` are related?

See page 22 of slides about generic types.

Name:	Number:

Part III -- XML (1 mark)

5 – [1.0 marks] Consider the problem presented in Part I-1.a). Present a DTD and XML document, with XML elements and attributes, to store all information needed. Use dummy values in the XML.

Part IV -- Swing (1 mark)

6 – [1.0 marks] What are the differences between Swing and AWT?

```
1) AWT component are considered to be heavyweight while Swing component are
lightweights.
2) Swing has plug-gable look and feel.
3) AWT is platform depended same GUI will look different on different platform
while Swing is developed in Java and is platform dependent.
```