# Further questions

Once this exercise has been finished, answer the following questions:

# Which methods can be really invoked on the collection elements?

All methods of the Figure class or super class can be invoked on the collection.

# If Castle class has implemented void castle () method, could it be possible to invoke that method from a reference to the base class? Why?

# No, it cannot because that method is properly of Castle class.

# What should we have to do in order to be able to use the previous void castle () method from an object of Castle class that is pointed by a reference to Figure class?

One way could be creating void castle () method in the Figure class, and overwrite this method in the Castle subclass.

# What should we do to know exactly to which class belongs every object pointed by a reference to the base class?

With the reserved word “instance of”

1. Answer the following questions:

# Show the difference between a class and an object.

The class is something abstract and the object is something that we can communicate and use its methods

# Which steps are involved in the instantiation's process of an object? The steps for a creation of objects are:

Declaration, instantiation, initialization.

# How is an object instantiated in Java?

# Student student= new Student();

# Answer the following questions:

* **What is inheritance?**

# In object-oriented programming, inheritance is the mechanism of basing an object or class upon another object or class having similar implementation.

# How do you express in Java that one class inherits from another?

With the word “extends”

# Which methods of the superclass are visible from the subclasses?

For the subclass all the methods of the superclass are visible, except the abstract methods, this are implemented in sub classes.

# What is the meaning of method overriding?

Meaning that allows a subclass to provide a specific implementation of a method that is already provided by one of its super class.

# Which type are both instantiated objects (in options 1 and 2)?

Are instanced of type Rectangle and Square respectively.

# Which type is the variable that references them?

GeometricFigure.

# Which methods from superclass are visible from the subclass?

By inheritance all the methods of the superclass are seen from the subclasses except that methods that have of access modifier private

# Can you use the same variable as a reference for different types of figures? Why?

Yes I can everyclass is going to inherit from the super class GeometricFigure.