# Introduction in .Net Core -> Lab 3 (Florin Olariu && Dan Nastasa)

Prerequisites:

a) Create a Blank solution
b) Add a Class Library (.NET Core)
c) Add a Unit Test Project (.NET Core)
d) Add dependency between Test Project and Class Library

Note: The exercise is meant to learn how to build domain models and it we help us to:

- Learn how to design classes
- How to apply aggregation
- How to use/apply encapsulation
- How to use/apply inheritance
- How the Repository Pattern works
- How to manipulate collections

## Exercise:

1. Create a class called **Rectangle** and expose the following properties:
   a. **Id**
      **Width(double)**
      **Height(double)**
      **Color**
      **Transparency(percentage)**
   b. Create a class called **Circle** and expose the properties:
      **Id**
      **Radius(double)**
      **Color**
      **Transparency(percentage)**
   c. Expose the following behaviour, for both classes:
      **ComputeArea**
      **IsVisible**
      **Draw(virtual or abstract?) =>** should return **"Drawing" + Rectangle or "Drawing" + Circle.**
   d. Using the inheritance principle extract a base class called **Shape**.
   e. Explain the option for the Draw method
   f. Create unit tests to have 100% code coverage for all the classes.
2. Create JobInfo class and expose the following:
   a. **Id**
      **Title**
      **MinimumYearsOfExperience**
      **Requirements**
      **Salary**
   b. Create Class **JobInfoRepository**
   c. Populate a List of job infos via constructor(minimum 3 jobs)
   d. Expose and test the following behaviour
      i. **GetJobByTitle**(string jobTitle)

ii. **FindAllJobs**()<br>
iii. **AddJob**(JobInfo job)<br>
iv. **GetJobByPosition**(int position)<br>
v. **RemoveJobsWithTitle**(string jobTitle)<br>
vi. **GetJobsWithSalaryHigherThan**(int limit)<br>
vii. Create unit tests to have 100% code coverage.

Note:

1. All exercises are mandatory.
2. You will receive your points at the end of the lab.