# Objectives

* Studying JavaDoc of an open source project.
* Using Input space partitioning techniques to design test cases.
* Learn testing code coverage.
* Practice on mocking dependencies in unit testing.

# Logistics

* You should work on teams
* Teams consists of 3 members exactly in the SAME group.
* It is essential to be in the same group, since there are different classes to test for each group.
* Send to your TA if you can’t find a team
* Working in teams consists of more or less than 3 members is disallowed unless you got a permission from your TA.
* Team members cannot change their teams in Phase 2, choose wisely because you are STUCK TOGETHER.

# Description

* In this assignment you will study the open source code of "JFreeChart" that we have gone through some of its classes in the lab.
* The JavaDoc is quite good for this code, and you are to study those documentations to be able to do the required work. You can find those docs in --> <http://www.jfree.org/jfreechart/api/javadoc/index.html>
* Group L3-S1 is to study the following classes (org.jfree.data.***ComparableObjectItem***, org.jfree.data.***ComparableObjectSeries***and org.jfree.data.general.***Series****)*
* Group L3-S2 to study the following classes (org.jfree.data ***DefaultKeyedValue***, org.jfree.data. ***DefaultKeyedValues***, and org.jfree.data.general.***KeyedValueComparator****)*
* Group L3-S3 to study the following classes (org.jfree.data ***KeyedObject***, org.jfree.data. ***KeyedObjects***, and org.jfree.data.***KeyedObjects2D****)*
* ALL groups should also the following classes (org.jfree.data.xy.***MatrixSeries*** and org.jfree.data. ***DataUtilities***) and you will find to test these two you will have to study also Values2D and KeyedValues, but don't test those, just study them to be able to mock them to test the two classes mentioned above.
* Once you study those API documentations, you will get to know the responsibilities behind those classes and the relations between them. Now you are required to use ***input space partitioning along with boundary values analysis***.
* Deliverables of phase 1:
  1. A report of all the test cases you have developed for the assigned classes mentioned above, with full reasoning of the partitions, the boundary values, and the chosen coverage criteria with complete explanation. In the written report, you should discuss how you are designing the test cases. This report can be a word document, or a set of compressed scanned pictures. Create your test-cases on paper (your written report) first.
  2. A testing package, with a test class for each class mentioned above (please follow java coding/naming standards).
  3. the zip file that you will upload to acadox that contains both the report and the test project, should be named as such --> stud1ID-stud2ID-stud3ID.zip
* The assigned classes should be covered with both ***unit tests*** and ***integration tests***(since you will find that, many of those classes have dependencies).
* Things to consider
  1. Since many of the assigned classes have dependencies, you may need to ***MOCK*** those dependencies to develop TRULY independent unit tests.
  2. You can use [JMock](http://jmock.org) or [mockito](https://site.mockito.org) or any other java mocking library if you please.
  3. You can use JUnit or [TestNG](https://testng.org/doc/) or any other java unit testing library if you please.
  4. Failing reasons for each unit test case should specifically be one reason.
  5. You can add your test package/classes to the included project, or you can create a new empty project and just add the jar files of JFreeChart (found in the "lib" directory in the included project)
  6. If you don't mock dependencies in your unit tests, marks will be deducted.
  7. Make the test design document as detailed as possible.
  8. Fail to name the zip file as mentioned above, marks will be deducted.
  9. Cheaters will get negative marks.

# Policy Regarding Plagiarism:

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر