# Project: University Administration System

**Cop 3331 – Object Oriented Software Design – Fall 2016 – USF**
**Due: 11-21-2016 @ 11:30 pm.**

**Instructor: José J Galvis** **Points: 100**

## 1. MAIN OBJECTIVE

The goal of this project is to design and implement a university administration system.

The system consists of classes of these types:

- Person (Student, Teacher)
- Student (B.Sc., M.Sc., Ph.D., ...)
- Teacher (lecturer, adjunct, professor, ...),
- Course.
- Department.

You can have additional classes if required.

## 2. DESCRIPTION

The system should have these relationships included:

- A Person should have the basic information of a person like: university ID, name, birth date, and gender.
- A student can be an undergraduate or graduate student.
- A student can enroll in a course.
- A graduate student can be a teaching assistant or research assistant.
- A teaching assistant can be assigned to a course.
- A teacher can be a lecturer, adjunct or professor.
- A teacher can be assigned to a course.
- Courses can be from undergraduate or graduate level.
- Courses should include the grades of all students enrolled in the class.
- Each teacher, student or course is assigned to a department.

## 3. TEST PROGRAM

The program should be tested using auxiliary files.

- Teachers.txt
- Students.txt
- Courses.txt
- Departments.txt

Each file should contain instances of the classes in the project.

The input to the system is obtained by reading these text files.

Each file should include enough instances to completely test the system's flow of events:

- Display information about Teachers
- Display information about Students
- Display information about Courses
- Display information about Departments

# 4. GRADING

The grading will mainly be based on these factors:

- How you use the concepts you have learned in this class (Object Oriented Design)
  - Inheritance,
  - Polymorphism,
  - Composition,
  - C++ structures, etc.
- The hierarchical design of the project using a UML class diagram,
- Well designed and structured code,
- Comments in the code.

# 5. ACADENMIC INTEGRITY

For this project, two students may work on the same project. The code must be 100% original. Code from any other party is not allowed in your project.

# 6. DELIVERABLES

Convert your UML class diagram to a PDF file.

Design your classes using both the interface (header) file and the implementation (.cpp) file.

Compile and link your classes and the client application.

Place all the required files: PDF for UML class diagram, source, object, executable and data (,h, .cpp, .o, .exe and .txt) into a folder and name the folder with your full name and Project (ex: JoeDoeProject).

Place the folder into a .zip file (ex: JoeDoeProject.zip) and upload the .zip file to Canvas by 11/21/16 @ 11:30 pm.