**CS 525 Software Testing and Quality Assurance**  
Department of Computer Science

**Lectures**  
R 06:20 PM - 08:50 PM, HT 235

**Instructor**  
Dr. G. Dimitoglou  
E-mail: dimitoglou[at]hood.edu  
Office Hours: Tue-Thu 5:00-6:00pm or by appointment.  
Office: HT 261

**Course Description**  
This course examines the theory and practice behind software testing and quality assurance. Emphasis is placed on understanding the software testing process, planning, strategy, criteria, and testing methods, as well as software quality assurance concepts & control process. Topics will include test models, test design techniques (black box and white-box testing), integration, regression, measurement, unit testing, slicing and debugging, inspection, and software metrics. Emerging concepts and their impact on testing will also be examined. This is both a theoretical and hands-on course. Multiple software testing suites will be used during the semester to enforce student mastery of the material.  
Prerequisite: CS 524 or permission of the instructor.

**Course Objectives**  
By the end of the course students will:

- Be able to testing software programs to software requirements using a test plan.  
- Understand and perform software cost estimation.  
- Understand software metrics and how they relate to testing  
- Understand and be able to apply various software testing  
- Know how to write and understand test plans and test procedures.  
- Be able to write a test plan based on the requirements document.  
- Explain problems and benefits in structured programming and machine language.  
- Understand the concepts of correctness and completeness as they relate to software quality

**Text(s)**  
Lecture notes and papers will be provided to students as needed.

**Assessment**  
Mid-Term (30%), Final Exam (35%), Homework (35%)
Topics (tentative)

1. Introduction and general principles
   a. Importance of software testing
   b. Definitions and principles.
   c. Terminology, quality and testing maturity model (TMM).
2. Black-box test techniques, Reliability and Usability
   a. Random testing, equivalence partitioning, boundary value analysis and error guessing, cause-and effect graphing and state-transition testing.
   b. Operational profiles, usage models, statistical testing, reliability, reliability (growth) models and usability.
   c. Statistical testing/usage-based testing and reliability growth models.
3. Inspections and white-box test techniques
   a. Control flow testing and data flow testing. Important areas are cyclomatic complexity, loop testing and mutation testing.
   b. Software inspections, reading techniques
   c. Fault content estimations.
4. Lifecycle and documentation
   a. Unit, integration, system and acceptance testing.
   b. Goals, policies, plans and documentation.
5. Organization and tools
   a. Advantages and disadvantages of test organizations.
   b. Skills, test group, roles and integration of testing activities.
   c. Evaluation of tools, tool categories, test automation.
6. Metrics and TMM
   a. Testing process monitor and control
   b. Test completion and configuration management.
   c. Testing maturity model (TMM). process improvement, assessment