

## School of Computing and Information Sciences

**Course Title:** Fundamentals of Software Testing

**Date:** 2/12/2018

**Course Number:** CEN 4072

**Number of Credits:** 3

|   |   |
|---|---|
| <b>Subject Area:</b> Software Engineering   | <b>Subject Area Coordinator:</b><br>Monique Ross<br><b>email:</b> moross@cs.fiu.edu |
| <b>Catalog Description:</b><br>Fundamentals of software testing. Topics include: test plan creation, test case generation, program inspections, specification-based and implementation-based testing, GUI testing, and testing tools. |   |
| <b>Textbook:</b> Aditya P. Mathur. "Foundations of Software Testing" 2014 Edition 2, Pearson, ISBN 9788131794760.   |   |
| <b>References:</b> The Art of Software Testing, Second edition. Glenford J. Myers. Revised and updated by Tom Badgett and Todd M. Thomas, with Corey Sandler. John Wiley and Sons, New Jersey, U.S.A., 2004. ISBN: 0-471-46912-2      |   |
| <b>Prerequisites Courses:</b> COP 3530  |   |
| <b>Corequisites Courses:</b> Includes a closed lab component  |   |

Type: Elective for Computer Science; Required for Software Design and Development Track

Prerequisites Topics:

- Significant Java or C++ programming experience

Course Outcomes:

1. Be familiar with creating, evaluating and implementing a test plan for a medium-size code segment.
2. Be familiar with program inspections.
3. Master the techniques used to perform specification-based testing and implementation-based testing on programs.
4. Be familiar with the techniques that apply test adequacy coverage criteria to the implementation model.
5. Be familiar with GUI testing.
- ~~6. Be exposed to program debugging.~~
- ~~7.6.~~ Be familiar with tools to support testing, and coverage analysis ~~and debugging.~~

**School of Computing and Information Sciences**  
**CEN 4072**  
**Software Engineering I**

**Relationship between Course Outcomes and Program Outcomes**

| <b>BS in CS: Program Outcomes</b>   | <b>Course Outcomes</b> |
|---|------------------------|
| a) Demonstrate proficiency in the foundation areas of Computer Science including mathematics, discrete structures, logic and the theory of algorithms             | 3, 4                   |
| b) Demonstrate proficiency in various areas of Computer Science including data structures and algorithms, concepts of programming languages and computer systems. | 3, 4                   |
| c) Demonstrate proficiency in problem solving and application of software engineering techniques  |                        |
| d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.   |                        |
| e) Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.   |                        |
| f) Demonstrate the ability to work cooperatively in teams.  | 1, 2                   |
| g) Demonstrate effective communication skills.  | 1                      |

**Assessment Plan for the Course & how Data in the Course are used to assess Program Outcomes**

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan:  
<http://www.cis.fiu.edu/programs/undergrad/cs/assessment/>

**School of Computing and Information Sciences**  
**CEN 4072**  
**Fundamentals of Software Testing**

**Outline**

| <b>Topic</b>  | <b>Number of<br/>Lecture Hours</b> | <b>Outcome</b> |
|---|------------------------------------|----------------|
| <ul style="list-style-type: none"> <li>• Introduction               <ul style="list-style-type: none"> <li>○ Testing process</li> <li>○ Limits of testing</li> <li>○ Importance of testing tools</li> </ul> </li> </ul>   | 4                                  | 1,3,4,7        |
| <ul style="list-style-type: none"> <li>• Test Plan               <ul style="list-style-type: none"> <li>○ Creation</li> <li>○ Evaluation</li> <li>○ Implementation</li> </ul> </li> </ul>   | 6                                  | 1              |
| <ul style="list-style-type: none"> <li>• Specification-based testing               <ul style="list-style-type: none"> <li>○ Equivalence class testing</li> <li>○ Boundary value testing</li> <li>○ State-transition testing</li> <li>○ Domain analysis testing</li> </ul> </li> </ul> | 8                                  | 3,4,7          |
| <ul style="list-style-type: none"> <li>• Implementation-based testing               <ul style="list-style-type: none"> <li>○ Program inspections</li> <li>○ Control flow testing</li> <li>○ Data flow testing</li> </ul> </li> </ul>  | 8                                  | 2,3,4,7        |
| <ul style="list-style-type: none"> <li>• GUI testing               <ul style="list-style-type: none"> <li>○ Capture playback</li> <li>○ Creating test scripts</li> <li>○ Using data pools</li> <li>○ GUI testing tools</li> </ul> </li> </ul>   | 5                                  | 5,7            |
| <ul style="list-style-type: none"> <li>• Debugging               <ul style="list-style-type: none"> <li>○ Debugging techniques</li> <li>○ Tools to support debugging</li> </ul> </li> </ul>   | 5                                  | 6,7            |

**School of Computing and Information Sciences**  
**CEN 4072**  
**Fundamentals of Software Testing**

**Course Outcomes Emphasized in Laboratory Projects / Assignments**

|   | <b>Outcome</b>   | <b>Number of Weeks</b> |
|---|--|------------------------|
| 1 | Software Test Document<br>Outcomes: 1,3,5,7                  | 2                      |
| 2 | Performing specification-based testing<br>Outcomes: 3,6,7    | 2                      |
| 3 | Performing implementation-based testing<br>Outcomes: 3,4,6,7 | 2                      |
| 4 | Performing GUI testing<br>Outcomes: 5,6,7                    | 2                      |

**Oral and Written Communication**

Number of written reports: 2 (Software Test Documents)

Approximate number of pages for each report: 30

Number of required oral presentations: 2 (Summary of test document and demonstration of tools to support testing)

Approximate time for each presentation: 15 minutes per group (5 minutes per student)

**Social and Ethical Implications of Computing Topics**

No significant coverage

**Approximate number of class hours devoted to fundamental CS topics**

| <b>Topic</b>                                   | <b>Core Hours</b> | <b>Advanced Hours</b> |
|--|-------------------|-----------------------|
| <b>Algorithms:</b>                             |                   | <b>1.0</b>            |
| <b>Software Design:</b>                        |                   | <b>0.5</b>            |
| <b>Computer Organization and Architecture:</b> |                   |                       |
| <b>Data Structures:</b>                        |                   | <b>1.0</b>            |
| <b>Concepts of Programming Languages</b>       |                   | <b>0.5</b>            |
| <b>Other CS Topics:</b>                        |                   |                       |

**School of Computing and Information Sciences**  
**CEN 4072**  
**Fundamentals of Software Testing**

**Theoretical Contents**

| Topic | Class time |
|-------|------------|
|       |            |

**Problem Analysis Experiences**

1. 

|                                   |
|-----------------------------------|
| Control flow analysis of programs |
|-----------------------------------|
2. 

|                                |
|--------------------------------|
| Data flow analysis of programs |
|--------------------------------|

**Solution Design Experiences**

1. 

|                      |
|----------------------|
| Design of test cases |
|----------------------|

**The Coverage of Knowledge Units within Computer Science Body of Knowledge<sup>1</sup>**

| Knowledge Unit      | Topic  | Lecture Hours |
|---------------------|--|---------------|
| <a href="#">SE2</a> | Debugging.   | 4             |
| <a href="#">SE3</a> | Testing tools.   | 10            |
| <a href="#">SE6</a> | Test plan creation, evaluation and implementation. Specification-based and implementation-based testing, program inspections and testing object-oriented software. | 20            |

---

<sup>1</sup>See [http://www.acm.org/education/curric\\_vols/cc2001.pdf](http://www.acm.org/education/curric_vols/cc2001.pdf) for a description of Computer Science Knowledge units