

## **CS 570: Software Testing**

### **School of Technology & Computing**

3 Credits, Graduate Course  
Fall 2020

*Access to the Internet is required.*

*All written assignments must be in Microsoft-Word-compatible formats.*

*See the library's APA Style Guide tutorial for a list of resources that can help you use APA style.*

## Contact Information

Professional experience information for instructors is found under *Faculty Information* in the online course menu.

## Faculty Information

Contact information for instructors is found under *Faculty Information* in the online course menu.

**Email:** [first name] [last name]

**Phone:** [xxx-xxx-xxxx]

**Office Hours and Response Time:** [I am available through MS Teams Tuesday and Thursday nights between 7-8pm. I will respond within 24 hours. I will grade within 3 business days after the due date.]

**Bio:** (keep images under 300px wide)

## Course Description

This course looks at the theories and practical tools and techniques for the testing and validation of software. Testing includes unit, integration, regression and user acceptance testing using both black-box and white-box techniques. The course also covers developing and writing test cases, creating error reports, and tracking test status. Upon completion of the course the student should be comfortable designing and applying requirements to software systems testing.

## Course Resources

Required and recommended resources to complete coursework and assignments are found on the course [Reading List](#). Note: resources listed under "Required - Must Purchase" should be purchased from a vendor of the student's own choosing; resources listed under "Available from the Library" are available at no cost to students.

### Textbook:

- Ammann, P. & Offutt, J. (2016). Introduction to Software Testing (2nd Edition). Cambridge, UK: Cambridge University Press. (ISBN: 9781107172012)

### Reference

- Ammann, P. & Offutt, J. (2016). Introduction to Software Testing (2nd Edition). Retrieved from <https://cs.gmu.edu/~offutt/softwaretest/>
- Fain, Y. (2011). [Java Programming: 24-Hour Trainer](#). Wrox Press. (ISBN:9780470889640)
- Kurniawan, B. (2015). [Java: A Beginner's Tutorial](#) (4th Edition). Brainy Software Corp. (ISBN: 9780992133047)
- junit.org. [JUnit 5 User Guide](#). Retrieved from <https://junit.org/junit5/docs/current/user-guide/>

# Course Outcomes

As a result of this course, students will know or be able to do the following:

- Interpret a product specification in order to create a test specification.
- Work productively within a team environment to develop a risk analysis to prioritize code testing and identify opportunities for automated tests and create a test plan.
- Design and perform automated and non-automated testing, write descriptive bug reports, and summarize the work as a team.
- Communicate testing, reports, and summary of work to identified stakeholders.
- Reflect on the course experience and provide improvements to the testing process.

# Additional Information

Topics Covered:

1. Foundations of Software Testing
  - When Software Goes Bad
  - Goals of Testing Software
  - Level Test Plan Example Template
2. Model-Driven Test Design
  - Software Testing Foundations
  - Software Testing Activities
  - Testing Levels Based on Software Activity
  - Coverage Criteria
  - Model-Driven Test Design
3. Test Automation with JUnit
  - Software Testability
  - Components of a Test Case
  - A Test Automation Framework
4. Criteria-Based Test Design
  - Coverage Criteria Defined
  - Infeasibility and Subsumption
  - Advantages of Using Coverage Criteria
5. Criteria Coverage: Inputs Space Partitioning
  - Input Domain Modeling
  - Combination Strategies Criteria
  - Handling Constraints Among Characteristics
  - Extended Example: Deriving an IDM from JavaDoc
6. Criteria Coverage: Graph Coverage
  - Graph Coverage Criteria
  - Graph Coverage for Source Code
  - Graph Coverage for Design Elements
  - Graph Coverage for Specifications
  - Graph Coverage for Use Cases
7. Criteria Coverage: Logic Coverage

- Semantic Logic Coverage Criteria (Active)
- Syntactic Logic Coverage Criteria (DNF)
- Structural Logic Coverage of Programs
- Specification-based Logic Coverage
- Logic Coverage of Finite State Machines
- 8. Criteria Coverage: Syntax-Based Testing
  - Syntax-based Coverage Criteria
  - Program-based Grammars
  - Integration and Object-Oriented Testing
  - Specification-based Grammars
  - Input Space Grammars
- 9. Bug Reporting
- 10. Managing the Test Process
  - Requirements Analysis and Specification
  - System and Software Design
  - Intermediate Design
  - Detailed Design
  - Implementation
  - Integration
  - System Deployment
  - Operation and Maintenance
  - Implementing the Test Process

## Grading Scale

The grades earned for the course will be calculated using City University of Seattle's decimal grading system, found in the current University Catalog (<https://www.cityu.edu/catalog/>).

Grading rubrics with details on how each assignment will be graded are located under *Assignments* and/or in *My Grades* in the online course menu. Students should review the rubric for each assignment prior to completing their work in order to understand how it will be assessed.

| OVERVIEW OF REQUIRED ASSIGNMENTS            | % OF FINAL GRADE | POINTS  |
|---|------------------|---|
| The Muddiest Point (MP)                     | 5%               | 50 = 5 points * 10 modules  |
| Concept Test (CT)                           | 5%               | 50 = 5 points * 10 modules  |
| Discussion Board (DB)                       | 10%              | 100 = 10 points * 10 modules  |
| Hands-On Practice (HOP)                     | 20%              | 200 = 20 points * 10 modules  |
| Software Testing Exercise (STE)             | 30%              | 300 = 30 points * 10 modules  |
| Knowledge Check (KC)                        | 10%              | 100 = 10 points * 10 modules  |
| Team Project (TP) - Comprehensive Test Pass | 20%              | Proposal: 30 points<br>Progress: 70 points<br>Final Report: 70 points<br>Final PPT: 30 points<br>Subtotal: 200 points |

|              |             |                     |
|--------------|-------------|---------------------|
| <b>TOTAL</b> | <b>100%</b> | <b>1,000 points</b> |
|--------------|-------------|---------------------|

## Course Assignments and Grading

The instructor will provide grading rubrics that will provide more detail as to how this assignment will be graded.

### The Muddiest Point (MP)

Before class, students are required to submit the Muddiest Point (MP) activity. The purpose of this activity is to stimulate student engagement. The instructor uses the MP to assess how students understood the required readings. The instructor also uses the MP to customize the lecture scope to implement Just-in-Time Teaching (JiTT). The MP consists of writing a brief reflective essay ( $\leq 50$  words) identifying the most confusing part (i.e., the MP) of the content covered in the upcoming module. If a student understood all concepts, the student needs to explain the most exciting aspect. There is one multiple-choice question from the required reading to demonstrate that the student understood the required readings.

| <b>Criteria</b> | <b>% of Grade</b> |
|-----------------|-------------------|
| Participation   | 40%               |
| Writing         | 40%               |
| Correctness     | 20%               |
| <b>TOTAL</b>    | <b>100%</b>       |

### Concept Test (CT)

The instructor poses a problem based on key concepts of a lecture. After reflecting on the problem, students submit their response and the instructor review them without providing a correct answer. Students discuss their thought process and solution with a peer. Students then commit to an answer and re-submits their responses. Instructor reviews responses and thought processes with the correct answer.

| <b>Criteria</b> | <b>% of Grade</b> |
|-----------------|-------------------|
| Engagement      | 100%              |
| <b>TOTAL</b>    | <b>100%</b>       |

### Discussion Board (DB)

All classes are required to use the Discussion Board. Participation through DB is an integral part of this course. It is defined as active engagement in a discussion or other activity. Instructors will determine the type of activities and their due dates; moreover, different DB activities will have different substance and length guidelines. The instructor will provide specific instructions to students.

A student posts an answer to a weekly discussion topic in Discussion Board. The student also posts a response to two other students' posts by the end of each module. Comments and

questions should be clear and thoughtful, with correct grammar, spelling, and punctuation. The instructor will grade the quality of your discussion postings on both content and response.

Questions or comments specifically for the instructor should be emailed directly to the instructor or posted in the Question and Answer Forum. Students who want to talk with other students about issues unrelated to the discussion forums should use the Coffee Talk Forum.

Although the tone of your DB postings can be informal, your instructor will expect the content to be on a professional level. Your comments and questions for discussion should be clear and thoughtful, with correct grammar, spelling, and punctuation. As with written assignments, the quality of your discussion postings will be graded on both content and presentation.

| <b>Criteria</b> | <b>% of Grade</b> |
|-----------------|-------------------|
| Participation   | 50%               |
| Writing         | 50%               |
| <b>TOTAL</b>    | <b>100%</b>       |

### **Hands-on Practice (HOP)**

The instructor will assign Hands-on Practice exercises to a pair of students in class or individually online. Students pair up and practice exercises to learn specific programming languages, application programming interfaces (APIs), or tools related to the programming assignments or virtual labs.

| <b>Criteria</b>   | <b>% of Grade</b> |
|-------------------|-------------------|
| Practice Exercise | 80%               |
| Engagement        | 20%               |
| <b>TOTAL</b>      | <b>100%</b>       |

### **Software Testing Exercise (STE)**

The students must individually perform the programming exercise. Programs must be executable and robust. Non-executable programs will not receive any credits. Programs should deliver correct answers on all valid input and produce comprehensible error messages on invalid input. Programs also run correctly on all test data given within a reasonable amount of time. Students should write programs that are easy for other people to read.

| <b>Criteria</b>       | <b>% of Grade</b> |
|-----------------------|-------------------|
| Program Execution     | 40%               |
| User Requirement      | 40%               |
| Program Documentation | 20%               |
| <b>TOTAL</b>          | <b>100%</b>       |

## Knowledge Check (KC)

Weekly quizzes measure knowledge concepts acquired. Focus on the underlying principles and concepts rather than memorization to solve the quizzes.

| <b>Components</b> | <b>% of Grade</b> |
|-------------------|-------------------|
| Correctness       | 100%              |
| <b>TOTAL</b>      | <b>100%</b>       |

## Team Project (TP)

Each student can select his or her team that consists of three students. A group of fewer than three students requires the instructor's approval. Each team will use an instructor-approved topic relevant to the course.

The paper must be no less than 6-7 pages. We required students to use the paper template from [EDSIG/CONISAR](#), the international conference standard. *The instructor may recommend the best papers in this course to conferences with your team's approval. If necessary, the instructor may require more revisions after the course is over. However, the paper submission is optional and has nothing to do with your course grade.*

The instructor will provide students three report templates and one presentation template. The file name consists of team project number, team number, and the list of the team members. For example, "TP01 T03 Sam John Mark."

- TP01 for the proposal - "TP01 T0X Author1 Author2 Author3.docx"
- TP02 for the progress report - "TP02 T0X Author1 Author2 Author3.docx"
- TP03 for the final report - "TP03 T0X Author1 Author2 Author3.docx"
- TP04 for the final presentation slide - "TP04 T0X Author1 Author2 Author3.pptx"

As in any scholarly writing, students should not merely copy information from another author. Students should use evidence to support the contentions they have drawn from their findings and critically analyze related literature. In essence, each paper needs to be an analytical paper, not a summary of readings.

In addition, a team presentation slide is required.

- The presentation consists of 15+4 slides: 15 slides for content and 4 slides for cover, agenda, key reference, and Q&A.
- The PPT template is provided. Your team can change design and color for your team's purpose.
- If necessary, a presentation video (15 minutes) may be requested.
- If necessary, a demo video (a maximum of 1-2 minutes) may be requested. But the demo time should be included in the total presentation time (15 minutes).

Four submissions are required according to the following schedule:

- Proposal (1 page; 30 points) - Starting (Module 1) & Ending (Module 3)

- Progress Report (3-4 pages; 70 points; graded after the proposal has been submitted) - Starting (Module 4) & Ending (Module 7)
- Final Report (6-7 pages; 70 points; graded after the progress has been submitted) - Starting (Module 8) & Ending (Module 9)
- Final PPT (15+4slides, 30 points; graded after the final report has been submitted) - Starting (Module 8) & Ending (Module 10)

Students are expected to use the assigned readings, videos, and other materials throughout the quarter. Students will need to utilize additional sources that were not assigned by the professor. While stylized after an industry report, nonetheless, students are expected to employ APA formatting of citations, footnotes, and bibliography. Students must cite the sources of all ideas, facts, and information used that are not their own, even if they have put the information into their own words. Failure to do so is plagiarism, although the oversight is unintentional. To avoid plagiarism, check <https://library.cityu.edu/howto/apa-writing/avoid-plagiarism/>.

### **Project Description: Comprehensive Test Pass Report**

Student test teams will implement system tests using automated test tools. There will be 4 source code modules within the scope of the assignment to perform tests on. The instructor will provide three of the four testing source code modules. The teams will prepare a report on the results of their testing efforts. This involves some team decision making on your part as well as some good test engineering, test execution, and creativity.

The project is very challenging and puts you and your team in the role of an actual test engineering team. This project is a real-world testing simulation over a 2-3 week test sprint period where you will be:

- Defining test cases,
- Running tests,
- Automating tests,
- Reporting/summarizing your test execution results.

Included are test cases, a list of tests run, code coverage, errors found in the code, inability to test some code, and workarounds you had to perform because of these barriers and any bugs you find reporting in a reproducible form by your team.

Attached will be 3 Java source code modules. In addition, as a team select a small program (less than 250 lines of code) for your 4th module from the Github directory (<https://github.com/testmycode>) on a topic of interest. Then follow the instructions below:

1. Document and perform unit tests on the 3 modules provided. Do not complete unit testing on the 4th module you selected. As you perform unit tests, report your results in your test execution summary report.
2. Develop and run code coverage on all 3 given modules and the module you chose from GitHub. As a team, use test engineering decision making skills to determine whether to run Branch Coverage or Statement Coverage or both, but you must run overall code coverage at a minimum. Report your results in your test execution summary as well.

3. Create automated tests using JUnit that include at least 3 assertions and 2 faults for each source module you test.
4. Once you complete the test execution and automation task above, you will then report any bugs you find using your own reporting tool. You will also report on code coverage and any required workarounds. It is OK if the bug count is small and non-critical, you should still report the bugs you have found. You will not be measured on the number of bugs reported but on the scope of your work.

As a team, then write a comprehensive test pass report that documents your test cases, automated tests, code coverage, test execution scope, bug reports, as well as discuss what areas you chose not to test and any trade offs/priority testing scope decisions and work around you made.

Each team will prepare a 15 minute presentation based on the comprehension test pass report. Present any decisions you made about test strategy, performing your testing, automation, reporting, workarounds, final test results, and recommendations. For presentation, all team members need to be involved equally. A link to a demo video part should be included in the report and the presentation.

- In-class: Present during the scheduled day for presentations.
- Online: Get together and record your presentation using any tool. MS Teams is a good tool.

### **TP Report Rubric**

The student will provide a report formatted based on a template provided by the instructor. Students are required to improve the writing iteratively and incrementally every week. The revision will always happen during a quarter. Students will add new required sections to the existing paper every week.

The final report is the culmination of applied research and activities conducted throughout the quarter. The final report/paper provides a detailed problem and its solution likely to be encountered by a company or organization described in a case study supplied by the student.

| <b><i>Criteria</i></b> | <b><i>% of Grade</i></b> |
|------------------------|--------------------------|
| Structure              | 20%                      |
| Content                | 30%                      |
| Writing                | 30%                      |
| Reference              | 10%                      |
| Collaboration          | 10%                      |
| <b>TOTAL</b>           | <b>100%</b>              |

### **TP Presentation Rubric**

The student will report on the research outcomes, development, or other project efforts to an academically appropriate committee in a public forum. The nature of the presentation content will determine the specific makeup of the audience. The student will choose the format of the presentation, in consultation with the advisor. The layout and design must be appropriate and adequate to represent the outcomes of the effort. While students must make some form of a

visual presentation, the presentation of the results may include publishing in a refereed publication, publication in a trade or popular magazine or journal, broadcast in an appropriate medium, or, in exceptional cases, limited dissemination within a closed community.

Each presenter will have 15 minutes for presentation and 5 minutes for questions and answers. Each presenter must keep the total presentation time limit strictly.

| <b>Criteria</b>             | <b>% of Grade</b> |
|-----------------------------|-------------------|
| Structure                   | 20%               |
| Visual Presentation         | 30%               |
| Verbal Quality & Engagement | 30%               |
| Collaboration               | 20%               |
| <b>TOTAL</b>                | <b>100%</b>       |

## Course Policies

Course policies on topics such as Late Assignments, Participation, and Professional Writing are found under Course Information in the online course menu. Students are responsible for reviewing and applying these policies while enrolled in this course.

## University Policies

Students are responsible for understanding and adhering to all of City University of Seattle's academic policies. The most current versions of these policies can be found in the [University Catalog](#) that is linked from the CityU Web site.

### Title IX Statement

City University of Seattle and its faculty are committed to supporting our students and seeking an environment that is free of bias, discrimination, and harassment. If students have encountered any form of sexual misconduct (e.g. sexual assault, sexual harassment, stalking, domestic or dating violence), we encourage them to report this to the University. If a student speaks with a faculty member about an incident of misconduct, that faculty member must notify CityU's Title IX coordinator and share the basic fact of the experience. The Title IX coordinator will then be available to assist students in understanding all of the options and in connecting students with all possible resources on and off campus.

To view CityU's sexual misconduct policy and for resources, please visit the [Title IX](#) and [Campus Safety](#) pages in the my.cityu.edu portal.

### Religious Accommodations

Washington state law requires that City University of Seattle develop a policy for accommodation of student absences or significant hardship due to reasons of faith or conscience, or for organized religious activities. The University's policy, including more information about how to request an accommodation, is available in the University Catalog. Accommodations must be requested within the first two weeks of this course using the Religious

Accommodations Request Form found on the student dashboard in the my.cityu.edu student portal.

### **Academic Integrity**

Academic integrity in students requires the pursuit of scholarly activity that is free from fraud, deception and unauthorized collaboration with other individuals. Students are responsible for understanding CityU's policy on academic integrity and adhering to its standards in meeting all course requirements. A complete copy of this policy can be found in the [University Catalog](#) under *Student Rights and Responsibilities* on the page titled *Academic Integrity Policy*.

### **Attendance**

Students taking courses in any format at the University are expected to be diligent in their studies and to attend class regularly.

Regular class attendance is important in achieving learning outcomes in the course and may be a valid consideration in determining the final grade. For classes where a physical presence is required, a student has attended if they are present at any time during the class session. For online classes, a student has attended if they have posted or submitted an assignment. A complete copy of this policy can be in the [University Catalog](#) under *Student Rights and Responsibilities* on the page titled *Attendance*.

## **Support Services**

### **Disability Services Accommodations Statement**

Students with a documented disability who wish to request academic accommodations are encouraged to contact Disability Support Services to discuss accommodation requests and eligibility requirements. Please contact Disability Support Services at [disability@cityu.edu](mailto:disability@cityu.edu) or 206.239.4752 or visit the [Disability Support Services](#) page in the my.cityu.edu portal. Confidentiality will be observed in all inquiries. Once approved, information about academic accommodations will be shared with course instructors.

### **Library Services**

CityU librarians are available to help students find the resources and information they need to succeed in this course. Contact a CityU librarian through the [Ask a Librarian](#) service, or access [library resources and services online](#), 24 hours a day, seven days a week.

### **Smarthinking Tutoring**

CityU students have access to free online tutoring offered through Smarthinking, including writing support, from certified tutors 24 hours a day, seven days a week. Contact CityU's Student Support Center at [help@cityu.ed](mailto:help@cityu.ed) to request a user name and password.