Homework Assignment 6 (50 Points + 50 Points Extra Credit)
CSE 464, Fall 2025
CIDSE, Arizona State University

Due:

Part I : Wednesday Dec 3rd 11:59 pm
Part II (Extra Credit): Friday Dec 5th 11:59 pm

Objectives:

Part I (50 Points): Special considerations in web application testing, Software architectural considerations in client-server web applications towards better quality, using web driver to test web applications.

Part II (50 Points Extra Credit): Part II of the assignment helps you to reinforce the topics discussed in the class including

a) Object Oriented Application Testing
b) Using Junit to develop and run test cases
c) State Based Testing

Part I – Web Application Testing

Reading: Lecture notes and example programs/test cases posted under lecture notes related to web application testing.

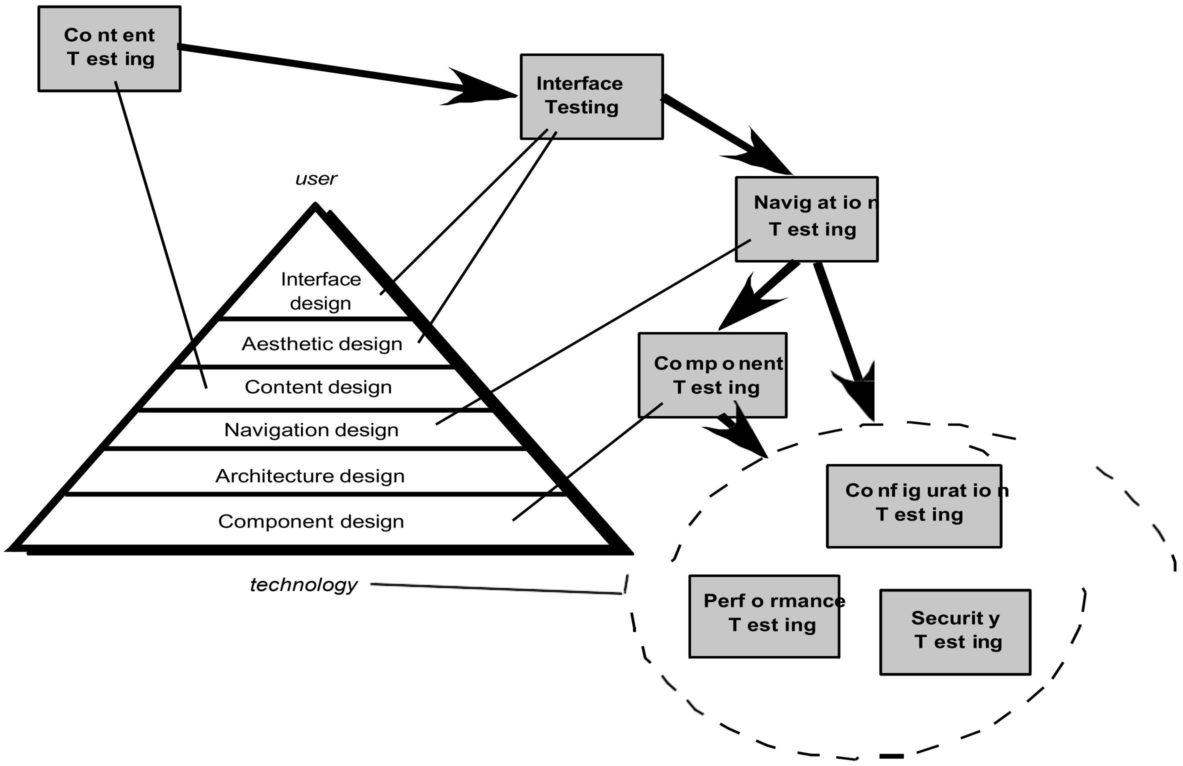
Pre-requisites: Configure Eclipse to run the Selenium web driver (see the tutorial posted in Module 11)

1.) [20 Points] **Testing Web Applications: Basic Concepts (Module 11 covers this topic)**

a) The gray box approach is the most appropriate in testing web applications. Briefly explain the difference between the gray box approach with respect to black box and white box testing.

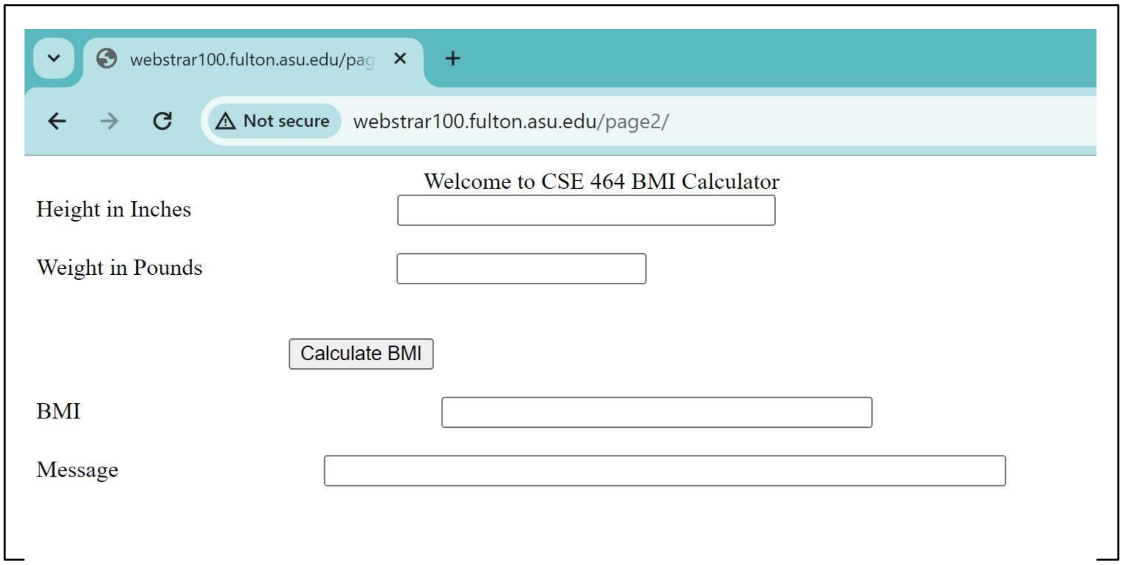
b) The following figure shows the mapping of web application design and testing (conventional 3-tire applications). Briefly explain testing considerations (what do we test and any tools that can be used) in each of the testing areas in the following figure.

Read web application testing notes posted with the homework.



2.**[30 Points] Using Selenium Toolkit:** Selenium is a toolkit that facilitates testing various aspects of web applications. Consider the following application that implements a simple BMI calculation web application **(Module 11 covers this topic)**

http://webstrar100.fulton.asu.edu/page2/


webstrar100.fulton.asu.edu/pag
Not secure
webstrar100.fulton.asu.edu/page2/
Calculate BMI

Welcome to CSE 464 BMI Calculator

Height in Inches

Weight in Pounds

BMI

Message

This web application computes the BMI using the following equation.

It calculates the BMI value using the following formula:

BMI = 703 * weight/(height)^2

Depending on the value of variables or BMI, the application also shows one of the following outputs along with BMI value:

Height or weight cannot be negative: height < 0 or weight < 0 Height or
weight cannot be zero: height = 0 or weight = 0 Underweight if BMI <
18.5
Normal weight if 18.5 <= BMI < 25 Overweight if
25 <= BMI < 30
Obese if BMI >= 30

Use Eclipse to develop selenium web driver-based test cases to test the BMI calculator web application. Your test cases should test the BMI value as well as the message to test if the correct BMI value is computed and the correct message is displayed.

Name your test program as bmiTest.java. Please use the ***Chrome driver***

Points will be given based on the correctness and the completeness of your test cases.

Submission: Use the homework 6 Part I submission template given to submit your solutions.

Part II – Object Oriented Application Testing

Reading: Module 13

Pre-requisites: Install Eclipse and configure Eclipse to run Junit test

Testing Object Oriented Applications

1.[15 Points]: Objective of this question is for you to revisit common terminologies and concepts use in software testing

a) In the class we discussed three basic approaches in Integration testing: Top down, bottom up and sandwich Explain each technique in detail highlighting pros and cons of each technique.

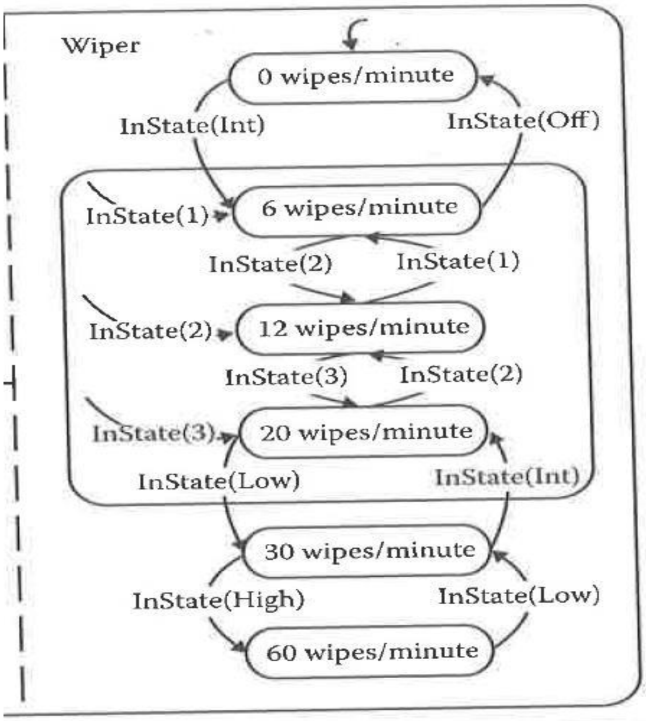
b) Junit provides four annotations @Before, @After, @BeforeClass, @AfterClass. Explain how you can use those testing OO applications (try to give examples)

2.[35 Pts State Based Testing] In the class we have discussed the windshield wiper program and developed test cases based on the StateChart for lever and a Use Case. In this question you will revisit the program design and develop two set of test cases.

Reading: The study the windshiledwiper program set and test cases posted before you attempt this question. Also read book section 15.3 windshield wiper example

StateChart Based Testing:

The following StateCart represent the behavior of the Wiper of the winshield wiper case study discussed in the class.

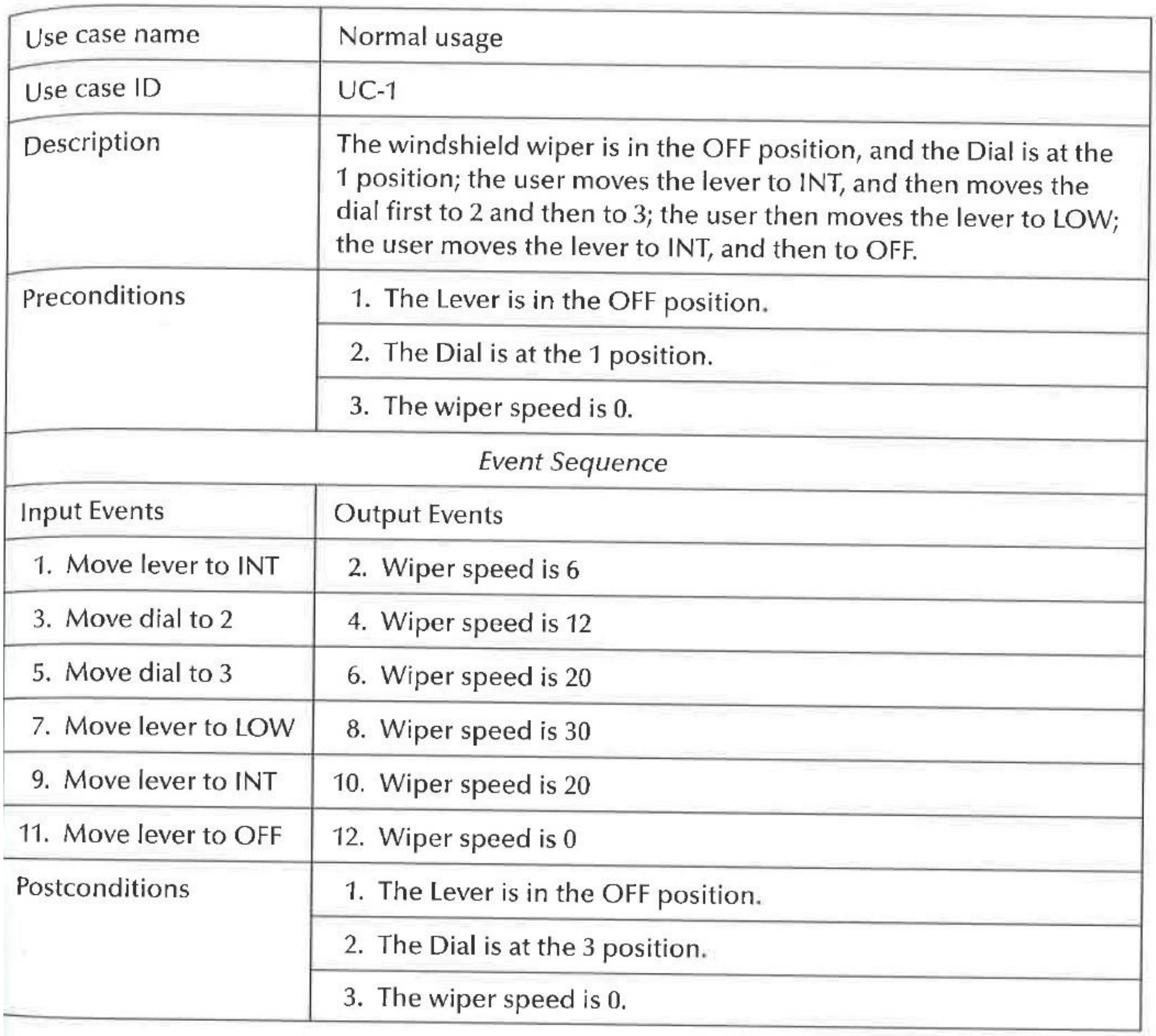


a) Design test cases based on the segment of the StateChart shown below. Your test cases should cover all state transitions shown in the given state diagram. Represent your test cases in a table.

b) Implement Junit test cases to implement your test cases designed above (a)

c) Use case based testing: Consider the following usage scenario.

Table 15.1 Lever and Dial Use Case



Use case name	Normal usage
Use case ID	UC-1
Description	The windshield wiper is in the OFF position, and the Dial is at the 1 position; the user moves the lever to INT, and then moves the dial first to 2 and then to 3; the user then moves the lever to LOW; the user moves the lever to INT, and then to OFF.
Preconditions	1. The Lever is in the OFF position.
	2. The Dial is at the 1 position.
	3. The wiper speed is 0.
Event Sequence	
Input Events	Output Events
1. Move lever to INT	2. Wiper speed is 6
3. Move dial to 2	4. Wiper speed is 12
5. Move dial to 3	6. Wiper speed is 20
7. Move lever to LOW	8. Wiper speed is 30
9. Move lever to INT	10. Wiper speed is 20
11. Move lever to OFF	12. Wiper speed is 0
Postconditions	1. The Lever is in the OFF position.
	2. The Dial is at the 3 position.
	3. The wiper speed is 0.

i.) Design test cases to test the scenario given.

ii.) Implement Junit test cases to implement your test cases designed above (c)

Assume that when lever and dial changes, they do not need to follow any sequential order.
For example, dial can go from 1 to 3 without staying at level 2.

Submission Instructions

Use the homework 6 Part II submission template given to submit your solutions