

CS163 Test Plan

Develop the test plan: For each member function that you plan to write, think about how to test it – what flow of control exists in the member function and how would you test out all conditions:

Test Case(s)	Expected Result	Verified? (yes/no)
CS-project-manager		
add-priority(1)	should add a list of priority 1	
remove-priority(1)	should remove list of priority 1	
remove-priority(non-existing)	should return FAILURE	
add-project(1, project)	should add project to the first priority list	
add-project(non-existing, project)	should return FAILURE	
remove-project(valid name)	should remove the corresponding project	
display-priority(valid priority)	should print out that priority	
display-priority(non-existing)	should return FAILURE	
display-all() → initialized list	should print everything	
display-all() → uninitialized list	should return FAILURE	
CS-project-list		
add-item(project)	should add the item to the list	
remove-item(valid name)	should remove the item	
remove-item(invalid name)	should return FAILURE	
display() → initialized list	should print everything	
display() → uninitialized list	should return FAILURE	

Verify correctness: Using the above test plan, create a test program that tests the interactions of all functions together.