

Software Testing		
<ul style="list-style-type: none"> ▪ Testing Setup ▪ Aspects of Testing ▪ Levels of Testing ▪ JUnit Framework ▪ JUnit Tests Example 		
Winter 2018	CISC124 – Section 2	1

Testing Setup		
<p>Core Activities in Testing Medium- to High Complexity-Software:</p> <ul style="list-style-type: none"> • Environment Setup → Hardware, Code Instrumentation, Test Automation Scripts • Data Preparation → Sets of Data Representing Special Events, Acceptable Values and Value, Ranges, Border Conditions • Verification of Assertions → Code-Oriented Assertions, Systemic Assertions, Software System Operation Assertions 		
Winter 2018	CISC124 – Section 2	2

Aspects of Testing		
<p>Core Aspects of Testing:</p> <ul style="list-style-type: none"> • Abstraction → Easily supports modifications to handle scaled up problems within a limited scope • Coverage → Capable to operate on different hardware and software with minimal modifications • Consistency → Easy to modularize and segregate components for testing and integration 		
Winter 2018	CISC124 – Section 2	3

Levels of Testing		
<ul style="list-style-type: none"> • Unit Testing → Scope is small sections of code under test for specific values, ranges and combinations of values • Integration Testing → Scope is interactions between sections of code (messages, functions calls, interaction protocols) and their insertion in the operational environment • User Acceptance Testing → Verification of correct responses produced by the software to high-level actions initiated by its users 		
Winter 2018	CISC124 – Section 2	4

Levels of Testing		
<ul style="list-style-type: none"> • Special Testing → Verifies specific aspects of the software mostly related to its operation <ul style="list-style-type: none"> – Safety Testing → Verifies the absence of unsafe events and states in the software – Load Testing → Verifies the conditions for acceptable performance levels in the operation of the software – Usability Testing → Verifies whether the software meets desired standards of operation and manipulation – Verification of Specifications Testing → Verifies that the software meets its original specifications. 		
Winter 2018	CISC124 – Section 2	5

JUnit Framework		
<ul style="list-style-type: none"> • SCOPE IS UNIT TESTING OF JAVA CLASSES AND THEIR METHODS <ul style="list-style-type: none"> – Environment → Requires a hosting environment (i.e. Eclipse) to easily assemble and run test cases. – Test Workflow → Assembled in one or more TEST CLASSES containing JUnit stubs that configure the testing actions to perform on one or more CLASSES UNDER TEST and their methods – Test Actions → Expressed in JUnit stubs that configure their sequencing or implement assertions about the code in the CLASS UNDER TEST. 		
Winter 2018	CISC124 – Section 2	6

JUnit Framework		
<ul style="list-style-type: none">• JUNIT STUBS:<ul style="list-style-type: none">– FLOW CONTROL<ul style="list-style-type: none">• @BeforeAll (Test method before a Class)• @AfterAll (Test method after a Class)• @BeforeEach (Test method before each Method)• @AfterEach (Test method after each Method)– ASSERTIONS<ul style="list-style-type: none">• assertEquals(), assertEqualsArray(), assertNotEquals() (for values or arrays of values)• assertSame(), assertNotSame() (for objects)• assertTrue(), assertFalse() (for boolean expressions)• assertNull() (for references)		
Winter 2018	CISC124 – Section 2	7

JUnit Test Example		
<ul style="list-style-type: none">• CLASS UNDER TEST: Student<ul style="list-style-type: none">– FOUR METHODS UNDER TEST• TEST CLASS: StudentTest<ul style="list-style-type: none">– IMPLEMENTS FLOW CONTROL– IMPLEMENTS VARIOUS TEST CASES FOR THE FOUR METHODS UNDER TEST		
Winter 2018	CISC124 – Section 2	8