

ENSF 380 - Group Project Submission 2

Test Name	Description	Test Value(s)
testSupplyConstructor	Creating a Supply object with a type and quantity should store both values correctly. getType() and getQuantity() are used to verify.	"Blanket", 5
testSetType	setType() should update the supply's type. getType() is used to check.	"Water" used in constructor, "Food" used by setType()
testSetQuantity	setQuantity() should update the supply's quantity. getQuantity() is used to check.	10 used in constructor, 20 used by setQuantity()
testSupplyQuantityZero	It should be possible to create a Supply with a quantity of 0.	"Blanket", 0
testSupplyQuantityNegative	Setting a negative quantity should throw an IllegalArgumentException.	"Blanket", -3
testInquirerConstructor	Creating an Inquirer with first name, last name, phone, and info should store all values. Getters are used to verify.	"John", "Doe", "403-555-1234", "looking for family"
testGetFirstName	getFirstName() should return the first name passed into the constructor.	"Jane"
testGetLastName	getLastName() should return the last name passed into the constructor.	"Smith"
testGetServicesPhoneNum	getServicesPhoneNum() should return the phone number passed into the constructor.	"587-111-2222"
testGetInfo	getInfo() should return the info string passed into the constructor.	"searching for son"
testInquirerEmptyFirstName	Creating an Inquirer with an empty first name should throw an IllegalArgumentException.	"", "Doe", "403-555-1234", "info"
testMedicalRecordConstructor	Creating a MedicalRecord with a Location, treatment details, and date of treatment should store all values. Getters are used to verify.	Location("Shelter A", "123 Main St"), "broken arm treatment", 2025-01-15

testSetLocation	setLocation() should update the medical record's location. getLocation() is used to check.	New Location("Hospital B", "456 Oak Ave")
testSetTreatmentDetails	setTreatmentDetails() should update the treatment details. getTreatmentDetails() is used to check.	"fracture treatment" used in constructor, "follow-up checkup" used by setter
testSetDateOfTreatment	setDateOfTreatment() should update the date of treatment. getDateOfTreatment() is used to check.	2025-01-15 used in constructor, 2025-02-20 used by setter
testMedicalRecordLocationObject	The Location object stored in a MedicalRecord should be the same object passed into the constructor. getName() on the Location is used to verify.	Location("Clinic C", "789 Elm St")
testMedicalRecordNullLocation	Creating a MedicalRecord with a null Location should throw an IllegalArgumentException.	null, "treatment", 2025-01-15
testMedicalRecordNullTreatment	Creating a MedicalRecord with null treatment details should throw an IllegalArgumentException.	Location("Shelter A", "123 Main St"), null, 2025-01-15
testFamilyRelationConstructor	Creating a FamilyRelation with two DisasterVictim objects and a relationship string should store all values. Getters are used to verify.	DisasterVictim("John", 2025-01-01), "sibling", DisasterVictim("Jane", 2025-01-01)
testSetPersonOne	setPersonOne() should update personOne. getPersonOne() is used to check.	New DisasterVictim("Alex", 2025-01-05)
testSetPersonTwo	setPersonTwo() should update personTwo. getPersonTwo() is used to check.	New DisasterVictim("Sam", 2025-01-05)
testSetRelationshipTo	setRelationshipTo() should update the relationship string. getRelationshipTo() is used to check.	"sibling" used in constructor, "parent" used by setter
testFamilyRelationPersonObjects	The DisasterVictim objects stored in a FamilyRelation should be the same objects passed in. getFirstName() on each person is used to verify.	DisasterVictim("Mike", 2025-01-01), DisasterVictim("Sara", 2025-01-01)

testFamilyRelationNullPersonOne	Creating a FamilyRelation with null as personOne should throw an IllegalArgumentException.	null, "sibling", DisasterVictim("Jane", 2025-01-01)
testFamilyRelationNullPersonTwo	Creating a FamilyRelation with null as personTwo should throw an IllegalArgumentException.	DisasterVictim("John", 2025-01-01), "sibling", null
testLocationConstructor	Creating a Location with a name and address should store both values. getName() and getAddress() are used to verify.	"Shelter A", "123 Main St"
testSetName	setName() should update the location name. getName() is used to check.	"Shelter A" used in constructor, "Shelter B" used by setName()
testSetAddress	setAddress() should update the location address. getAddress() is used to check.	"123 Main St" used in constructor, "456 Oak Ave" used by setAddress()
testAddOccupant	addOccupant() should add a DisasterVictim to the location's occupants. getOccupants() is used to verify the victim is in the array.	DisasterVictim("John", 2025-01-01)
testRemoveOccupant	removeOccupant() should remove a DisasterVictim from the location's occupants. getOccupants() is used to verify the victim is no longer in the array.	DisasterVictim("John", 2025-01-01) added then removed
testAddSupply	addSupply() should add a Supply to the location's supplies. getSupplies() is used to verify the supply is in the array.	Supply("Water", 10)
testRemoveSupply	removeSupply() should remove a Supply from the location's supplies. getSupplies() is used to verify the supply is no longer in the array.	Supply("Water", 10) added then removed
testSetOccupants	setOccupants() should replace the entire occupants array. getOccupants() is used to check the new array.	Array of 2 DisasterVictim objects (Victim A, Victim B)
testSetSupplies	setSupplies() should replace the entire supplies	Array of 2 Supply objects (Supply A, Supply B)

	array. getSupplies() is used to check the new array.	
testLocationOccupantsStartEmpty	A newly created Location should have an empty occupants array.	getOccupants().length == 0
testLocationSuppliesStartEmpty	A newly created Location should have an empty supplies array.	getSupplies().length == 0
testDisasterVictimConstructorTwoArgs	Creating a DisasterVictim with a first name and entry date should store both values. getFirstName() and getEntryDate() are used to verify.	"John", 2025-01-10
testDisasterVictimConstructorThreeArgs	Creating a DisasterVictim with first name, entry date, and date of birth should store all three values.	"Jane", 2025-01-10, 1990-05-20
testSetFirstName	setFirstName() should update the first name. getFirstName() is used to check.	"John" used in constructor, "Mike" used by setter
testSetLastName	setLastName() should update the last name. getLastName() is used to check.	"Doe"
testSetDateOfBirth	setDateOfBirth() should update the date of birth. getDateOfBirth() is used to check.	1990-05-20
testSetGender	setGender() should update the gender string. getGender() is used to check.	"male"
testSetComments	setComments() should update the comments. getComments() is used to check.	"needs wheelchair access"
testAddPersonalBelonging	addPersonalBelonging() should add a Supply to the victim's personal belongings. getPersonalBelongings() is used to verify.	Supply("Blanket", 2)
testRemovePersonalBelonging	removePersonalBelonging() should remove a Supply from the victim's personal belongings. getPersonalBelongings() is used to verify the supply is gone.	Supply("Blanket", 2) added then removed

testAddFamilyConnection	addFamilyConnection() should add a FamilyRelation to the victim's family connections. getFamilyConnections() is used to verify.	FamilyRelation with another DisasterVictim, "sibling"
testRemoveFamilyConnection	removeFamilyConnection() should remove a FamilyRelation from the victim's family connections. getFamilyConnections() is used to verify.	FamilyRelation added then removed
testAddMedicalRecord	addMedicalRecord() should add a MedicalRecord to the victim's medical records. getMedicalRecords() is used to verify.	MedicalRecord(Location("Shelter A", "123 Main St"), "flu treatment", 2025-01-12)
testSetPersonalBelongings	setPersonalBelongings() should replace the entire personal belongings array. getPersonalBelongings() is used to check.	Array of 2 Supply objects (Supply A, Supply B)
testSetFamilyConnections	setFamilyConnections() should replace the entire family connections array. getFamilyConnections() is used to check.	Array of 1 FamilyRelation object
testSetMedicalRecords	setMedicalRecords() should replace the entire medical records array. getMedicalRecords() is used to check.	Array of 1 MedicalRecord object
testGetEntryDate	getEntryDate() should return the entry date passed into the constructor. The entry date should not be changeable after construction.	2025-01-10
testDateOfBirthBeforeEntryDate	setDateOfBirth() should throw an IllegalArgumentException if the date of birth is after the entry date.	Entry date: 2025-01-10, date of birth: 2025-03-15
testSetFirstNameEmpty	setFirstName() with an empty string should throw an IllegalArgumentException.	""
testPersonalBelongingsStartEmpty	A newly created DisasterVictim should have an empty personal belongings array.	getPersonalBelongings().length == 0
testFamilyConnectionsStartEmpty	A newly created DisasterVictim should have	getFamilyConnections().length == 0

	an empty family connections array.	
testMedicalRecordsStartEmpty	A newly created DisasterVictim should have an empty medical records array.	getMedicalRecords().length == 0
testReliefServiceConstructor	Creating a ReliefService with an Inquirer, DisasterVictim, date, info, and Location should store all values. Getters are used to verify.	Inquirer("John", "Doe", "403-555-1234", "looking for son"), DisasterVictim("Mike", 2025-01-01), 2025-01-15, "last seen near river", Location("Shelter A", "123 Main St")
testSetInquirer	setInquirer() should update the inquirer. getInquirer() is used to check.	New Inquirer("Jane", "Smith", "587-222-3333", "looking for daughter")
testSetMissingPerson	setMissingPerson() should update the missing person. getMissingPerson() is used to check.	New DisasterVictim("Sara", 2025-01-05)
testSetDateOfInquiry	setDateOfInquiry() should update the date of inquiry. getDateOfInquiry() is used to check.	2025-01-15 used in constructor, 2025-02-01 used by setter
testSetInfoProvided	setInfoProvided() should update the info provided. getInfoProvided() is used to check.	"last seen near river" used in constructor, "found near bridge" used by setter
testSetLastKnownLocation	setLastKnownLocation() should update the last known location. getLastKnownLocation() is used to check.	New Location("Hospital B", "456 Oak Ave")
testGetLogDetails	getLogDetails() should return a string containing the inquirer name, missing person name, date of inquiry, info provided, and last known location.	Verify the returned string contains relevant details from all fields
testReliefServiceObjectReferences	The Inquirer, DisasterVictim, and Location objects stored should be the same objects passed in. Getters on each object are used to verify.	Check that getInquirer().getFirstName(), getMissingPerson().getFirstName(), getLastKnownLocation().getName() match original objects