Test Script should be in a table format, with header as shown below. There should be **at least 3 distinct test classes (as indicated in the description) **per function**.

Test descriptions are supposed to be unique and should indicate classes/ groups of test cases on what is being tested. For example, given the function getAreaTri() which computes the area of a triangle given the base and height as parameters, the following are 3 distinct classes of tests:

- testing with base and height values smaller than 1
- testing with whole number values for base and height
- testing with floating-point number values for base and height, larger than 1

The following test descriptions are incorrectly formed:
Too specific: testing with base containing 0.25 and height containing 0.75
Too general: testing if function can generate correct area of triangle
Not necessary: since already defined in pre-condition: testing with base or height containing negative values

FILL-UP THE FOLLOWING TABLE (the 1st three rows are just examples; delete it in your own document). Add new rows as you deem necessary...

Function Name	#	Test Description	Sample Input	Expected Result	Actual Result	Pass or Fail?
InitializeReset	1	pfRobotX, pfRobotY, and pdRobotAngle are positive numbers	pfRobotX = 100, pfRobotY = 100, pdRobotAngle = 100	pfRObotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRObotX = 0, pfRobotY = 0, pdRobotAngle = 0	Pass
InitializeReset	2	pfRobotX, pfRobotY, and pdRobotAngle are negative numbers	pfRobotX = -100, pfRobotY = -100, pdRobotAngle = -100	pfRObotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRObotX = 0, pfRobotY = 0, pdRobotAngle = 0	Pass
InitializeReset	3	pfRobotX, pfRobotY, and pdRobotAngle are zero	pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRObotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRObotX = 0, pfRobotY = 0, pdRobotAngle = 0	Pass
TranslateForward	1	fDistance and dRobotAngle are positive	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 45	pfRobotX = 70.7107..., pfRobotY = 70.7107...	pfRobotX = 70.7107..., pfRobotY = 70.7107...	Pass
TranslateForward	2	fDistance is zero and dRobotAngle is positive	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 45	pfRobotX = 0.0..., pfRobotY = 0.0...	pfRobotX = 0.0..., pfRobotY = 0.0...	Pass
TranslateForward	3	fDistance is negative and dRobotAngle is positive	fDistance = -100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 45	pfRobotX = -70.7107..., pfRobotY = -70.7107...	pfRobotX = -70.7107..., pfRobotY = -70.7107...	Pass

Function Name	#	Test Description	Sample Input	Expected Result	Actual Result	Pass or Fail?
TranslateForward	4	fDistance is positive and dRobotAngle is negative.	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = -45	pfRobotX = 70.7107..., pfRobotY = -70.7107...	pfRobotX = 70.7107..., pfRobotY = -70.7107...	Pass
TranslateForward	5	fDistance is zero and dRobotAngle is positive.	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = -45	pfRobotX = 0.0... pfRobotY = 0.0...	pfRobotX = 0.0... pfRobotY = 0.0...	Pass
TranslateForward	6	fDistance and dRobotAngle are negative.	fDistance = -100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = -45	pfRobotX = -70.7107..., pfRobotY = 70.7107...	pfRobotX = -70.7107..., pfRobotY = 70.7107...	Pass
TranslateForward	7	fDistance is positive and dRobotAngle is zero.	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRobotX = 100.0..., pfRobotY = 0.0...	pfRobotX = 100.0..., pfRobotY = 0.0...	Pass
TranslateForward	8	fDistance and dRobotAngle are zero.	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRobotX = 0.0..., pfRobotY = 0.0...	pfRobotX = 0.0..., pfRobotY = 0.0...	Pass
TranslateForward	9	fDistance is negative and dRobotAngle is zero.	fDistance = -100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRobotX=-100.0..., pfRobotY = 0.0...	pfRobotX=-100.0..., pfRobotY = 0.0...	Pass
TranslateBackward	1	fDistance and dRobotAngle is positive	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 45	pfRobotX = -70.7107..., pfRobotY = -70.7107...	pfRobotX = -70.7107..., pfRobotY = -70.7107...	Pass
TranslateBackward	2	fDistance is zero and dRobotAngle is positive	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 45	pfRobotX = 0.0..., pfRobotY = 0.0...	pfRobotX = 0.0... pfRobotY = 0.0...	Pass
TranslateBackward	3	fDistance is negative and dRobotAngle is positive	fDistance = -100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 45	pfRobotX = 70.7107..., pfRobotY = 70.7107...	pfRobotX = 70.7107..., pfRobotY = 70.7107...	Pass

Function Name	#	Test Description	Sample Input	Expected Result	Actual Result	Pass or Fail?
TranslateBackward	4	fDistance is positive and dRobotAngle is negative.	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = -45	pfRobotX = -70.7107..., pfRobotY = 70.7107...	pfRobotX = -70.7107..., pfRobotY = 70.7107...	Pass
TranslateBackward	5	fDistance is zero and dRobotAngle is negative.	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = -45	pfRobotX = 0.0..., pfRobotY = 0.0...	pfRobotX = 0.0... pfRobotY = 0.0...	Pass
TranslateBackward	6	fDistance is negative and dRobotAngle is negative.	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = -45	pfRobotX = 70.7107..., pfRobotY = -70.7107...	pfRobotX = 70.7107..., pfRobotY = -70.7107...	Pass
TranslateBackward	7	fDistance is positive and dRobotAngle is zero.	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRobotX=-100.0..., pfRobotY = 0.0...	pfRobotX=-100.0..., pfRobotY = 0.0...	Pass
TranslateBackward	8	fDistance and dRobotAngle are zero.	fDistance = 0, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRobotX = 0.0..., pfRobotY = 0.0 pfRobotX = 0.0..., pfRobotY = 0.0...	pfRobotX = 0.0, pfRobotY = 0.0	Pass
TranslateBackward	9	fDistance is negative and dRobotAngle is zero.	fDistance = 100, pfRobotX = 0, pfRobotY = 0, pdRobotAngle = 0	pfRobotX = 100.0..., pfRobotY = 0.0...	pfRobotX = 100.0..., pfRobotY = 0.0...	Pass
RotateCounter-Clockwise	1	dTheta is positive.	dTheta = 10, pdRobotAngle = 80	pdRobotAngle = 90.0...	pdRobotAngle = 90.0...	Pass
RotateCounter-Clockwise	2	dTheta is zero.	dTheta = 0, pdRobotAngle = 80	pdRobotAngle = 80.0...	pdRobotAngle = 80.0...	Pass
RotateCounter-Clockwise	3	dTheta is negative.	dTheta = -10, pdRobotAngle = 80	pdRobotAngle = 70.0...	pdRobotAngle = 70.0...	Pass
RotateClockwise	1	dTheta is positive.	dTheta = 10, pdRobotAngle = 80	pdRobotAngle = 70.0...	pdRobotAngle = 70.0...	Pass
RotateClockwise	2	dTheta is zero.	dTheta = 0, pdRobotAngle = 80	pdRobotAngle = 80.0...	pdRobotAngle = 80.0...	Pass
RotateClockwise	1	dTheta is negative.	dTheta = -10, pdRobotAngle = 80	pdRobotAngle = 90.0...	pdRobotAngle = 90.0...	Pass