Full Name:	Rana Mohamed Hamza Hassan Saleh.	Contact No:	1211708075			
Email ID:	ranamhsaleh@gmail.com	Category:	Student			
file:///C:/Users/Amr%20Mohamed%20Hamza/OneDrive/Documents/3rd%20year%202nd%20term/jMemorize-master/Software-Testing-and-Quality-Assurance-Assignment-1/Test%20Results(bugs%20report)%20%E2%80%94%20QuarterClassTest.html						
What is being tested:	Teat Case:	Input:	Expected Output:	Actual Output:	Status:	Screen Shot
Quarter()	calling Quarter()	Current year, month from the Calendar instance and in	Created a Quarter	Created a Quarter	Pass	testQuarterDefaultCtor
Quarter(java.util.Date)	calling Quarter(java.util.Date)	Date 1-1-2003	Constructs a new Quarter, based on a date/time and the defaul	Constructs a new Quarter, based on a date/time and the def	Pass	testQuarterDateCtor
Quarter(java.util.Date , java.util.TimeZ	Quarter(java.util.Date , java.util.TimeZone)	Date, TimeZone 1-1-100 "GMT"	Constructs a Quarter, based on a date/time and time zone.	Constructs a Quarter, based on a date/time and time zone.	Pass	testQuarterDateAndTimeZoneCtor
	calling Quarter(int, int) with valid input	int quarter, int year 1, 2003	Constructs a new quarter. based on the quarter and the year	Constructs a new quarter. based on the quarter and the yea	Pass	testQuarterIntIntCtor
	calling Quarter(int, int) with QuarterLessThan 1	int quarter, int year -1, 2003	IllegalArgumentException	did not throw an exception	Fail	testQuarterIntIntCtor_QuarterLessThan1
	calling Quarter(int, int) with Quarter Greater Than 4	int quarter, int year 5,2003	IllegalArgumentException	did not throw an exception	Fail	testQuarterIntIntCtor_QuarterGreaterThan4
	calling Quarter(int, int) with Year Greater Than 9999	int quarter, int year 2,10000	IllegalArgumentException	IllegalArgumentException	Pass	testQuarterIntIntCtor_YearGreaterThan9999
Quarter(int, int)	calling Quarter(int, int) with Year Less Than 1900	int quarter, int year 2,1899	IllegalArgumentException	IllegalArgumentException	Pass	testQuarterIntIntCtor_YearLessThan1900
	calling Quarter(int ,Year) with valid input	int quarter, Year year 1, 2003	Constructs a new quarter.	Constructs a new quarter.	Pass	testQuarterIntAndYearCtor
	calling Quarter(int ,Year) with Quarter Less Than 1	int quarter, Year year -1, 2003	IllegalArgumentException	did not throw an exception	Fail	testQuarterIntAndYearCtor_QuarterLessThan1
	calling Quarter(int ,Year) with Quarter Greater Than 4	int quarter, Year year 5, 2003	IllegalArgumentException	did not throw an exception	Fail	testQuarterIntAndYearCtor_QuarterGreaterThan4
	calling Quarter(int ,Year) with Year Less Than 1900	int quarter, Year year 1, 1899	IllegalArgumentException	IllegalArgumentException	Pass	testQuarterIntAndYearCtor_YearLessThan1900
Quarter(int ,Year)	calling Quarter(int ,Year) with Year Greater Than 9999	int quarter, Year year 1, 10000	IllegalArgumentException	IllegalArgumentException	Pass	testQuarterIntAndYearCtor_YearGreaterThan9999
getQuarter()	calling getQuarter()	1, 2003 to get the quarter	The quarter of the "Quarter" object.	The quarter of the "Quarter" object.	Pass	testGetQuarter
getYear()	calling getYear()	1, 2003 to get the year	The year of the "Quarter" object.	The year of the "Quarter" object.	Pass	testGetYear
	calling previous()	1, 2001 to get the previous	The "Quarter" preceding the current one (4, 2000)	The "Quarter" preceding the current one (4, 2000)	Pass	testPrevious
	Year Is 1900 And Quarter Less Than Or Equal 1	1, 1900 to get the previous	NULL	NULL	Pass	testPrevious_YearIs1900AndQuarterLessThanOrEqual1
previous()	Year Is 1900 And Quarter Greater Than 1	2, 1900 to get the previous	The "Quarter" preceding the current one (1,1900)	The "Quarter" preceding the current one (1,1900)	Pass	testPrevious_YearIs1900AndQuarterGreaterThan1
	calling next()	4, 2001 to get the next	The "Quarter" following the current one(1, 2002)	The "Quarter" following the current one(1, 2002)	Pass	testNext
	Year Is 9999 And Quarter Less Than Or Equal 3	3, 9999 to get the next	The "Quarter" following the current one (4, 9999)	The "Quarter" following the current one (4, 9999)	Pass	testNext_YearIs9999AndQuarterLessThanOrEqual3
next()	Year Is 9999 And Quarter Greater Than 3	4, 9999 to get the next	NULL	NULL	Pass	testNext_YearIs9999AndQuarterGreaterThan3
getSerialIndex()	calling getSerialIndex()	1, 2000 to get the serial index	The Serial index numver for the "Quarter" (2000*4+1)	The Serial index numver for the "Quarter" (2000*4+1)	Pass	testGetSerialIndex
	Same Quarter And Same Year	another object to test equality against (1, 2001)(1, 2001	TRUE	TRUE	Pass	testEquals_SameQuarterAndSameYear
	Different Quarter And Same Year	another object to test equality against (1, 2001)(2, 2001	FALSE	FALSE	Pass	testEquals_DifferentQuarterAndSameYear
	Same Quarter And Different Year	another object to test equality against (1, 2000)(1, 2001	FALSE	FALSE	Pass	testEquals_SameQuarterAndDifferentYear
	Different Quarter And Different Year	another object to test equality against (1, 2001)(2, 2002	FALSE	FALSE	Pass	testEquals_DifferentQuarterAndDifferentYear
equals()	Non Quarter Object	another object to test equality against (1, 2000)(non qu	FALSE	FALSE	Pass	testEquals_NonQuarterObject
hashCode()	calling hashCode()	quarter1(2, 2003) quarter2(2, 2003)	TRUE since both of them are equal	TRUE since both of them are equal	Pass	testHashCode
	Next Quarter	anotherobject to comape against (1, 2003)(2, 2003)	TRUE	TRUE	Pass	testCompareTo_NextQuarter
	Previous Quarter	anotherobject to comape against (2, 2003)(1, 2003)	TRUE	TRUE	Pass	testCompareTo_PreviousQuarter
compareTo(java.lang.Object)	Equal Quarter	anotherobject to comape against (2, 2003)(2, 2003)	TRUE	TRUE	Pass	testCompareTo_EqualQuarter
toString()	calling toString()	(1, 2000)	TRUE since both of them are equal	TRUE since both of them are equal	Pass	testToString
getFirstMillisecond(java.util.Calendar	calling getFirstMillisecond(java.util.Calendar)	A "Calender" object to determine the time zone (1, 200	TRUE since both of them are equal	TRUE since both of them are equal	Pass	testGetFirstMillisecond
getLastMillisecond(java.util.Calendar)	calling getLastMillisecond(java.util.Calendar)	A "Calender" object to determine the time zone (1, 200	TRUE since both of them are equal	TRUE since both of them are equal	Pass	testGetLastMillisecond
	Dash Delimiter Quarter Year	A string representing the quarter (Q2-2024)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_DashDelimiterQuarterYear
	Space Delimiter Quarter Year	A string representing the quarter (Q2 2024)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_SpaceDelimiterQuarterYear
	Forward SlashDelimiter Quarter Year	A string representing the quarter (Q2/2024)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_ForwardSlashDelimiterQuarterYear
	Comma Delimiter Quarter Year	A string representing the quarter (Q2,2024)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_CommaDelimiterQuarterYear
	Space Delimiter Year Quarter	A string representing the quarter (2024 Q2)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_SpaceDelimiterYearQuarter
	Forward SlashDelimiter Year Quarter	A string representing the quarter (2024/Q2)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_ForwardSlashDelimiterYearQuarter
	Comma Delimiter Year Quarter	A string representing the quarter (2024,Q2)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_CommaDelimiterYearQuarter
	Dash Delimiter Year Quarter	A string representing the quarter (2024-Q2)	TRUE since its valid	TRUE since both of them are equal	Pass	testParseQuarter_DashDelimiterYearQuarter
parseQuarter(java.lang.String)	Invalid Format	A string representing the quarter other than the previo	TimePeriodFormatException	TimePeriodFormatException	Pass	testParseQuarter_InvalidFormat