Test Plan – for reference

Class: WorkshopPaper

ID	Aim	Inputs	Expected Output	Planning or Code Check
P.D.1	Default constructor : Test that the constructor sets the title and creates an array of reviews	Paper title<String> Array of Reviews <Object> (number <= 3)	Object with the paper title and array of reviews	Planning
P.D.2	Default constructor :Test if a default object is made for empty paper title	Empty string for paper title Array of Reviews (number <= 3)	Object with default the paper title and array of reviews	Planning
P.D.3	Default constructor	Empty string for paper title No reviews	Object with default the paper title and empty array of reviews	Planning
P.D.4	Default constructor	Paper title<String> No reviews	Object with the paper title and empty array of reviews	Planning
P.M.1	Main Constructor : Test for when the input is correct , the paper title is set .	<String> Paper title	Paper title is set	Planning
P.M.2	Main Constructor : Check when the input is empty an error is printed	Empty string	Error message	Planning
P.GT.1	GetPTitle: Test the string is correctly returned	nothing	String	Code Check

P.ST.1	SetPTitle: Test the title is set properly	<String> paper title	Class paper title variable set with the string	Planning
P.ST.2	SetPTitle: Test error message is thrown if the string is empty	Empty string	Error message	Planning
P.AR.1	AddReview: Test review is added correctly	Review object	Object added to array	Planning
P.AR.2	AddReview: Test that appropriate error is thrown when input exceeds maximum amount in array	Review object when already 3 reviews in array	Error message	Planning
P.AR.3		No review object/ null	Error message	Planning
P.AS.1	Test the avg score is correctly created	Review object average score	Float for the average score	Planning
P.tS.1	Test the toString correctly creates the	Paper object	Average of reviews and each review on separate lines	Planning

Class: WorkshopReview

ID	Aim	Inputs	Expected Output	Planning or Code Check
1	Default Constructor: create the review object, with a default low score, and a default empty review.		the review object (a default low score, and a default empty review)	Planning
2	Main Constructor: create the object with these two values (should contain a reasonable minimum amount of text)	a score as an integer and a review as a string	the object with these two values (should contain a reasonable minimum amount of text)	Planning
3	Get Score method: returns the score for that review		the score for that review	Planning
4	Set Score method: sets the value to the correct variable. It returns nothing, but produce an error if something is wrong.	a score as an integer	produce an error if something is wrong.	Planning
5	Get Review method: returns the text-based review part as a String		the text-based review part of the object, as a String	Planning
6	Set Review method: Allows for the update of the text-based review, it should produce an error if something is wrong.	taking a string as an input	produce an error if something is wrong.	Planning
7	*toString()* method: Overrides the standard *toString()* method		printing out the score for the paper, the text-based review on a new line.	Planning

3) Class Test Plan

Class: Workshop System Main

Function:

ID	Aim	Inputs	Expected Output	Planning or Code Check
1.1	Offer several options, enter into or exit a loop and call different functions based on user input	User input(based on options)	Messages(asking for input, error)	Planning
1.2		User input(seeing an overview)	Print one line overview of all papers, if paper doesn't exists then print error message.	Code check
1.3		User input(adding a paper)	asking user to input title of paper, if nothing wrong print confirmation message else error messages)	Code check
1.4		User input(add a review to	Asking user to choose paper, then	Code check
		paper)	asking input a score. If nothing wrong print confirmation message else error message.	

1.5		User input(see detail about paper)	Asking user to input paper id, then, print paper id, detailed overview	Code check
1.6		User input(exit)	Exiting message	Code check
2	Create a workshop paper object, adds it to array of papers,	Title of paper	Message(asking for input, confirmation, errors)	Planning
3	Create a new review object , add it to paper	Paper(ask user to choose) User input the score	Message(confirmation, error)	Planning
4	Print one line overview of all papers in workshop		Print one line overview of all papers, Message(error)	Planning
5	Print paper id and detailed overview of specified paper	Paper id	Paper id, detailed overview *Paper.toString(),* message(error)	Planning

2) Bug / Fix List

ID	Problem Description	Proposed Fix	Priority High, Med, Low
Workshop review.2	This function doesn't contain a reasonable minimum amount of text	Set a minimum amount and check if the input is lower than the amount then print out error message and asking user to reinput.	Med
Worksop review.6	Doesn't have the ability to print error message if something goes wrong.	Set a criteria to decide if something is wrong. If something is wrong print error message.	High
Workshop paper.7	Doesn't need to return. It cannot be change once String type variable is being defined. Doesn't print out the desired results	Print average score and three reviews directly using *System.out.println(),* and do the calculation inside the print function.	High
Workshop paper.1	Copy and pasted code	Combine WorkshopPaper into one method that checks if the input is empty. For example: with an if else condition at the beginning of the method that sets default if empty string and sets paper title otherwise	Med
Workshop paper. 2	Have many repetitive codes	Change the codes to a loop, the stratified functions if and else can join in an 'if else' loop that add one	Med
		condition in each loop, and add a stop range in the end.	
Workshop paper.3		Combine the three if blocks into one for loop which loops through the array and tests if the current review is null and does the appropriate calculation based on that	Med
Workshop paper .4	No error messages	Add error checking and error messages to functions	High