

<p><i>Place candidate's barcoded sticker here.</i></p> <p><i>This sheet must be stapled to the front of each candidate's submission.</i></p>		I.T. PRACTICAL EXAMINATION 2017			
		Version: FINAL Marksheet (25 October 2017)		Prog. Lang.	
		Candidate:			
		Marker:		120	0
		Checksum: 00-0000-0000-0120		Total	Pupil
		Database Used:			
SQL					
1.1	SELECT * FROM tblProfiles ORDER BY DailyPostRate DESC				
1.2	SELECT Username, Fullname FROM tblProfiles WHERE Privacy="public" (LIKE "public" is acceptable but do not accept LIKE "public*" or variations of wildcards. Also for Q1.7)				
1.3	INSERT INTO tblMembership (GroupID, ProfileID, Admin, Posts) VALUES (8, 98, TRUE, 0); (Not necessary to award or deduct a mark for field list as it is not needed.)				
1.4	SELECT GroupID, ROUND(AVG(Posts), 1) FROM tblMembership GROUP BY GroupID;				
1.5	SELECT MONTH(CreatedDate) AS CreatedMonth, COUNT(*) AS TotalGroups FROM tblGroups GROUP BY MONTH(CreatedDate) HAVING COUNT(*) > 2;				
1.6	SELECT GroupID SUM(DailyPostRate * 7) AS PredictedWeeklyPosts FROM tblProfiles, tblMembership WHERE tblProfiles.ProfileID = tblMembership.ProfileID GROUP BY GroupID (Accept SUM(DailyPostRate) * 7)				
1.7	SELECT GroupName, COUNT(*) FROM tblProfiles, tblGroups, tblMembership WHERE tblProfiles.ProfileID = tblMembership.ProfileID AND tblGroups.GroupID = tblMembership.GroupID AND Admin = true AND Privacy = "private" (See 1.2) GROUP BY GroupName;				
1.8	UPDATE tblGroups SET Icon = LEFT (Icon, LEN(Icon) – 5) & ".jpg" WHERE RIGHT(Icon, 4) = ".jpeg";				
Programming					
2.1	Post Class: Class header is correct				
2.2	Fields/Properties: attributes declared private name correctly and correct type				
2.3	Constructor: header is correct with correct parameters and values assigned to correct attributes				
2.4	Accessor Methods (getPostID): named correctly, return type is correct, and returns the correct attribute value (Accept getID() as method name)				
2.5	toString: method header correct correct attributes returned with the correct formatting				

3.1	Response Class: Class header is correct		
3.2	Fields/Properties: attributes declared private name correctly and correct type (Do not give mark for correct type if responseType is not an int. Do not penalise for extra unrelated fields)		
3.3	Constants: final/const named correctly integer values assigned correctly (Do not penalise for private instead of public)		
3.4	Constructor: method header correct inResponseType parameter is an integer (penalise this again even if penalised in 3.2) values assigned to correctly attributes		
3.5	getResponseTime/getReactionType: method header correct (accept both names) switch or if...else used to compare against constants correct string returned (-1 mark if constants aren't used. -1 mark if four separate if statements that are used with no else) (Do not penalise for public instead of private)		
3.6	Accessor Methods (getPostID): named correctly, return type is correct, and returns the correct attribute value (Do not accept getID() in this case)		
3.7	toString: method header correct correct attributes concatenated and getResponseType used comments return different string to other responses (Do not penalise again for not using constants in the if statement)		
4.1	InstaPageManager: Class header is correct		
4.2	Properties: Private array of Post objects and array of Response objects declared correctly and both of correct size (100 and 500 respectively) Two counters declared and initialised to 0		
4.3	Constructor: Method header correct. Display error if file does not exist. Open file for reading. Loop through every line in the file. Read a line from the file and tokenize. Determine whether the line is a Post or Response . Extract the responseType for Response objects. Add a new Response object or a new Post object to the correct array at the correct position and increment the correct counters.		
4.4	getAllPosts: method header correct, appropriate for loop, concatenate the toString of each object with a newline and string returned		
5.1	Create an interface/form/class/unit called InstaPgeUI		
5.2	Instantiate a InstaPageManager Object		
5.3	getAllPosts method called and output displayed		
6.1	PostWithResponse Class: Class header is correct		
6.2	Fields/Properties: Property for a Post object OR inherited fields from Post class. Property declared for an array of Response objects (Additional fields, e.g. counters, are acceptable)		
6.3	Constructor: one parameter is a Post object and the other parameter is any array of Response objects Post parameter assigned to the correct property OR parent constructure called and array of Response objects assigned correctly (Additional parameters e.g. counters, are acceptable)		
6.4	toString: Post object toString method called OR super.toString() called loop through array of Response objects add a new line with some tabs included and concatenate each Response object to the output and return a string		
7.1	Array of PostWithResponse objects declared of the same size or larger size of the Post object array Method declared in InstaPageManager with an appropriate name that returns a string Loop through existing Post array Array declared to store the Response objects for a single post regardless of size. i.e. can include a counter or use Vector or ArrayList Loop through all Response objects and find all the matching Response objects for each Post (using the PostID and ID fields of the Response and Post objects respectively) Add the matching Response objects for each Post to an array of Responses Create a new PostWithResponse object and add to the array of PostWithResponse objects Concatenate and return the toString of each PostWithResponse object Some algorithm for ensuring that the array of Response objects associated with each Response is of the correct size (also accept including a counter, Vector or ArrayList or checking for null either here or in PostWithResponse class)		
7.2	Method called in InstaPageUI and output displayed		