Specific marking instructions

Task	Expected response		Additional guidance	Marks available	
1	**Software design and development**				
1a	1 mark each for: ♦ Input inside loop ♦ Conditional loop used ♦ Correct loop conditions		Condition could be implemented in a variety of ways.	3	Design (3)
1b	Initial Inputs	♦ Starting miles ♦ Number of charging stations		1	Implementation (15)
	Two fixed loops, each for number of charging stations entered			1	
	Input Validation (rating)	Conditional loop with correct condition	Valid inputs are 7, 22, 50	1	
		Input of rating within loop	Award 1 mark if implemented without input validation loop	1	
		Error message displayed inside loop		1	
	If statement	If structure matches design (else if of nested ifs)	Assignments 7 – 0 22 - 0.005 50 – 0.01	1	
		If conditions correct		1	
		Price per mile assigned correctly		1	
	Calculations	Calculate and store miles travelled in an array	Only miles travelled data should be stored in the array	1	
		♦ Input of currentMiles in loop ♦ Store new startmiles		1	
		Calculate and store cost of each journey stage in an array	Only journey stage data should be stored in the array	1	
	Both running totals calculated correctly within the second loop			1	
	Display each journeyStage cost			1	
	Display total stage cost rounded to 2 decimal places			1	
	Display total miles with message		Concatenation is not required	1	

Task	Expected response	Additional guidance	Marks available	
1	**Software design and development**			
1ci	Printed evidence of test run showing correct output	**Output:** (Stage 1 cost =) **0.6** (Stage 2 cost =) **0.91** (Total cost =) **1.51** Total miles = **211** Note that message for total miles may change. The first three outputs do not require a message.	1	Testing (3)
1cii	One mark each for: ♦ Journey stage costs ♦ Total miles and Total cost	Journey stage 1 cost = **0** Journey stage 2 cost = **-5.5** Total cost = **-5.5** Total miles = **-200**	2	
1ciii	One mark for: The miles at each stage should be validated to ensure its larger than the previous mileage.		1	Evaluation (4)
1d	Evaluation of the following for: (Efficiency) 1 mark: ♦ One efficiency or one inefficiency in own program code (Robustness) 1 mark: ♦ Program is robust or not, including example from own program code (Readability) 1 mark: ♦ Rreadability — comment on one aspect of readability in the candidate's own code	Efficiency examples could include comparison of: ♦ array vs multiple variables ♦ nested ifs vs individual ifs ♦ use of a loop vs replication of code Robust examples might refer to: ♦ input validation of kw rating ♦ lack of validation for other inputs ♦ current mileage potentially being incorrect Evaluation of readability must contain an element of evaluation rather than simple statements of terms. For example "I have used white space to highlight structures in my program" not "I have used white space". The candidate's code must also show evidence of this for a mark to be awarded.	3	