National 5
Coursework
Assessment Task

National 5 Computing Science
Assignment
Finalised Marking instructions

Marking instructions

General marking principles

Always apply these general principles. Use them in conjunction with the specific marking instructions, which identify the key features required in candidates' responses.

a Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted for errors or omissions.

b If a candidate response does not seem to be covered by either the principles or detailed/specific marking instructions, and you are uncertain how to assess it, you must seek guidance from your team leader.

c Award marks regardless of spelling, as long as the meaning is unambiguous and does not result in a syntax error in implemented code.

d For design and implementation tasks, a sample response may be shown in the detailed marking instructions. This will not be the only valid response. You must use the detailed marking instructions and additional guidance to ensure that you consider alternative approaches and nuances of different programming languages. If in doubt, you should refer to your team leader.

e A correct response can be negated if the candidate includes an extra, incorrect response which demonstrates they do not know the correct answer. For example, in a 'state' question where the only correct answer is 'white', if the candidate answers 'white orange', you should not award the mark.

f If a candidate puts a score through their entire response to a question and makes a further attempt, you should only mark the further attempt. If no further attempt is made and the original is legible, you should mark the original response.

g In the marking instructions, if a word is underlined then it is essential; if a word is in brackets() then it is not essential. Words separated by / are alternatives.

Specific marking instructions

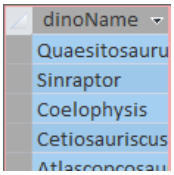
Task 1 — software design and development

Task	Expected response	Max mark	Additional guidance
1a	♦ Inputs (both required) – duration of training session – duration of each song ♦ Process (any 3 required) – add up the duration of all songs – compare total to the duration of training session (is greater than or equal to) – select a random song – calculate the length of the training session to seconds – validate length of training session	**2**	
1b	♦ running total within loop ♦ loop terminating when total length of songs is greater or equal to than training session duration ♦ output message (enough songs have been entered)	**3**	Make sure the condition matches the type of loop (pre or post conditional). If candidate fails to indicate a loop only the 3rd bullet can be awarded.
1c	♦ 3, 2:300, 3:400 ♦ foam message in correct place	**2**	Correct order: ♦ Number of songs 3 ♦ 2: 300 ♦ foam message ♦ 3:400

Task	Expected response	Max mark	Additional guidance
1d	**Step 2** (3 marks) ♦ conditional loop with correct condition ♦ input of training session duration within loop ♦ error message displayed inside loop **Step 3** (1 mark) ♦ calculate and store training session duration in seconds **Step 4** (5 marks) ♦ Initialize songCounter, increment songCounter in loop ♦ conditional loop continues iteration while total<duration of training session ♦ each song duration stored in array ♦ calculate running total in loop ♦ using if condition total >= duration of training session, in loop, to display enough songs message **Step 5** (6 marks) ♦ display number of songs with message ♦ store random number equal to number of songs entered ♦ fixed loop for correct number of songs ♦ display each song number with colon and song duration, within second loop ♦ if statement used to display foam message when randomly selected song number is reached ♦ display total (song duration) with message	**15**	Valid inputs are >=10 and <=30 Award 1 mark if implemented without loop Ensure condition matches the type of loop used. Could loop for length of array. Example format: 1:300 It must be possible for any of the songs to be selected. An alternative solution could make use of the loop variable.

Task	Expected response	Max mark	Additional guidance
1e	Efficiency (2 marks) ♦ two comments on efficiency or inefficiency of own code Robustness (1 mark) ♦ comment on one aspect of robustness of own code	**3**	Candidates require more than "My 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 for duration of training session ♦ lack of validation for other inputs

Task 2 — database design and development

Task	Expected response	Max mark	Additional guidance
2a	♦ FossilID ♦ countryFound, year, person	2	Candidates may choose their own input names to represent those shown. The only acceptable additional input, for the second bullet, is the FK.
2b	♦ Type and Required columns ♦ (Range) 0.1-50 ♦ (Restricted choice) Creataceous, Jurassic, Triassic	3	Type column values should be: – number – text – text Required should all be Y
2c(i)	♦ `INSERT INTO Fossil (fossilID, countryFound, year, person, dinoID)` ♦ `VALUES (2061,'USA', 2023,'Garath','DINO_68' );`	2	Also accept: ♦ `INSERT INTO Fossil` ♦ `VALUES (2061, 'USA', 2023, 'Garath', 'DINO_68');` Do not award any marks if the candidate's SQL is created by an application.
2c(ii)	♦ `SELECT dinoName FROM Dinosaur, Fossil` ♦ `WHERE countryFound="USA" AND year>1980` ♦ `AND fossil.dinoID = dinosaur.dinoID` ♦ `ORDER BY year (ASC);` WHERE clause when generated using MS Access design view: `WHERE (((Fossil.[countryFound])="USA") AND ((Fossil.[year])>1980) AND ((Fossil.dinoID)=[dinosaur].[dinoID]))`	4	Do not penalise candidates for additional fields in SELECT. Do not award any marks if the candidate's SQL is created by an application. First few rows are: 

Task	Expected response	Max mark	Additional guidance
2c(iii)	♦ `UPDATE Dinosaur SET length=8.0, diet = "Herbivorous"` ♦ `WHERE dinoName = "Aardonyx";`	**2**	Also allow use of primary key: `WHERE dinoID = "DINO_01"` Do not award any marks if the candidate's SQL is created by an application.
2d	♦ country field does not exist ♦ Only shows 1930 dinosaurs ♦ No equi join	**2**	Award 1 mark for each bullet. Maximum 2 marks. Allow answers that give solutions: ♦ field should be countryFound ♦ the year should be between 1930 and 1939 ♦ a join should be included

Task 3 — web design and development

Task	Expected response	Max mark	Additional guidance
3a	♦ display the name of the toy 'Hamish the Highland Coo'. ♦ display the description of the toy. ♦ display information about the original price (£19.99) ♦ display information about the sale price (£5.99) ♦ display an image of the toy ♦ toy image must be 250x400 pixels ♦ include a link to the Scotland Wikipedia page. ♦ include a link back to the homepage.	**2**	Award 1 mark for each bullet. Maximum 2 marks. Some detail required so "display image", "display text" would not be clear enough for a mark.
3b	♦ Wireframe split into four areas: – heading section – content section – links section – bottom banner (graphic) ♦ Content – name of the toy (hamish the highland coo) – toy description – original price information (£19.99) – sale price information (£5.99). – image of the toy. – link to the scotland wikipedia page. – link back to the homepage.	**2**	

Task	Expected response	Max mark	Additional guidance
3c	♦ heading - "Hamish the Highland Coo" ♦ an image of the toy cow 250px by 400px. ♦ a description of toy and information about the original price (£19.99) and sale price (£5.99). ♦ an external link to the Scotland Wikipedia page and a hyperlink back to the homepage.	**4**	Marks for should be awarded from CSS and HTML code, not the printout of webpage as viewed in a browser.
3d	The video 'highlandCow.mp4' has been added correctly.	**1**	Marks for should be awarded from CSS and HTML code, not the printout of webpage as viewed in a browser. Position and size of video doesn't matter.
3e	♦ edit external style sheet to: – colour contentSection white: (#FFFFFF) – colour linksSections white: (#FFFFFF) – colour hyperlinks Bright orange (#FFAC1C) ♦ style only the janPromotion page, content section text: – size: 18pt – text colour: Plum (#DDA0DD) – text Alignment: centre ♦ Sale price size: 24pt	**3**	Marks for should be awarded from CSS and HTML code, not the printout of webpage as viewed in a browser. If the text on the whole janPromotion page is styled (for example using <body>) award marks for styling the content section. Sale price could be styled using an ID, class or inline style.
3f	♦ matches user-interface design ♦ links and navigation work correctly ♦ media (such as text, graphics, and video) display correctly ♦ consistency	**2**	Award 1 mark for each bullet. Maximum 2 marks.
3g	website is fit/not for purpose	**1**	Answer must relate to candidate's own code completed in 3c and 3d.

[END OF MARKING INSTRUCTIONS]

Candidate: **Total marks awarded:** **/40**

Task 1 – Software Design and Development

		Marks Available	Marks Awarded	
1a – Analysis	Inputs x 2	1		
	Processes x 3	1		**/2**
1b – Design	Running total within loop	1		
	Loop terminating	1		
	Output message	1		**/3**
1c – Testing	3, 2:300 and 3:400	1		
	Foam message in correct place	1		**/2**

1d – Implementation:

Step 2	Conditional loop with correct condition	1		
	Input of training session within loop	1		
	Error message displayed inside loop	1		
Step 3	Calculate and store training session in seconds	1		
Step 4	Initialise and increment songCounter in loop	1		
	Conditional loop (total < duration)	1		
	Each song duration stored in array	1		
	Calculate running total in loop	1		
	Display enough songs message using condition in loop	1		
Step 5	Display number of songs with message	1		
	Store random number equal to number of songs entered	1		
	Fixed loop for number of songs	1		
	Display each song number : song duration	1		
	Use if to display foam message	1		
	Display total (song duration) with message	1		**/15**
1e – Evaluation	Efficiency x 2	2		
	Robustness	1		**/3**

Task 2 – Database Design and Development

		Marks Available	Marks Awarded	
2a – Analysis	FossilID	1		
	countryFound, year, person	1		/2
2b – Design	Type and required columns	1		
	Range check	1		
	Restricted choice	1		/3
2c(i) – Implementation	INSERT statement	1		
	VALUES statement	1		/2
2c(ii) – Implementation	SELECT, FROM	1		
	WHERE clause (USA, > 1980)	1		
	Join	1		
	ORDER BY year (ASC)	1		/4
2c(iii) – Implementation	UPDATE statement	1		
	WHERE clause	1		/2
2d – Testing	Reason 1	1		
	Reason 2	1		/2

Task 3 – Web Design and Development

		Marks Available	Marks Awarded	
3a – Analysis	Functional requirement 1	1		
	Functional requirement 2	1		/2
3b – Design	Wireframe spilt into four areas	1		
	Content added	1		/2
3c – Implementation	Heading	1		
	Image added 250x400	1		
	Description and prices of toy	1		
	External link and home link	1		/4
3d - Implementation	Video added	**1**		/1
3e – Implementation	Colours implemented x 3	1		
	Only janPromotion page styled	1		
	Sale price: 24pt	1		/3
3f – Testing	Test 1	1		
	Test 2	1		/2
3g – Evaluation	Fitness for purpose	**1**		/1