



Unit: Introduction to Programming

Assignment title: Capital Cruises

Sample Marking Scheme

Markers are advised that, unless a task specifies that an answer be provided in a particular form, then an answer that is correct (factually or in practical terms) **must** be given the available marks. If there is doubt as to the correctness of an answer, the relevant NCC Education materials should be the first authority.

This marking scheme has been prepared as a **guide only** to markers and there will frequently be many alternative responses which will provide a valid answer.

Each candidate's script must be fully annotated with the marker's comments (where applicable) and the marks allocated for each part of the tasks.

Throughout the marking, please credit any valid alternative point.

Where markers award half marks in any part of a task, they should ensure that the total mark recorded for the task is rounded up to a whole mark.

Marker's comments:

Moderator's comments:

Mark:

Moderated mark:

Final mark:

Penalties applied for academic malpractice:

Design

	Guide	Maximum Marks
1	<p>Design the visual aspect of the forms:</p> <ul style="list-style-type: none"> • Choice of controls (to a maximum of 4 marks) <ul style="list-style-type: none"> ○ 4 marks: Suitable components, neatly laid out. Use of combo boxes where appropriate to minimise screen real estate and suitable method for selection of item options to prevent invalid input. ○ 2 - 3 marks: Suitable components, neatly laid out. Components ensure all choices available but may not be the most efficient choices in terms of screen real estate or preventing invalid input. ○ 1 mark: Some required components missing. Poor GUI layout. ○ 0 marks: Not attempted, or no suitable controls chosen for input choices. 	4 marks
	<p>Suitable choice of colours and fonts: (to a maximum of 4 marks)</p> <ul style="list-style-type: none"> ○ 4 marks: All colours and fonts suitable and consistently applied. ○ 2 - 3 marks: Either suitable, consistently applied colour scheme or fonts, but not both. ○ 1 mark: Suitable fonts or colours but lacking consistency. ○ 0 marks: Colours of controls and fonts used make text hard to read. 	4 marks
	<p>Naming conventions (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> ○ 2 marks: Naming conventions followed for all controls. ○ 1 mark: Naming conventions followed for some controls. ○ 0 marks: Naming conventions not followed. 	2 marks
1		Total 10 marks
2	<p>Pseudocode for calculating the cost (to a maximum of 12 marks):</p> <ul style="list-style-type: none"> • Loop over the room type combo box items to find which item is selected (2 marks max – 1 mark for correct for loop syntax and number of repetitions, 1 mark for the correct if statement criteria to identify whether the current item is selected) • Access the cost of the room type selected from the room prices array from the appropriate element (1 mark) • Add the cost of the selected room type to the total variable (1 mark) • Loop over the food package combo box items to find which item is selected (2 marks max – 1 mark for correct for loop syntax and number of repetitions, 1 mark for the correct if 	

	Guide	Maximum Marks
	<p>statement criteria to identify whether the current item is selected)</p> <ul style="list-style-type: none"> • Access the cost of the room type selected from the food package prices array from the appropriate element (1 mark) • Add the cost of the selected food package type to the total variable (1 mark) • Check if the shore option has been selected and, if it has, add the shore cost to the total (1 mark) • Check if the spa option has been selected and, if it has, add the shore cost to the total (1 mark) • Check if the fitness option has been selected and, if it has, add the shore cost to the total (1 mark) • Check if the internet option has been selected and, if it has, add the shore cost to the total (1 mark) 	<u>12 marks</u>
	<p>Pseudocode for validating the town name (to a maximum of 4 marks)</p> <ul style="list-style-type: none"> • Check whether the length of the text in the town/ city textbox is fewer than 2 characters in length (1 mark) • If the length is less than 2 characters, display a message asking to re-enter (1 mark) • Check whether the length of the text in the town/ city textbox is greater than 25 characters in length (1 mark) • If the length is greater than 25 characters, display a message asking to re-enter (1 mark) 	<u>4 marks</u>
	<p>Complete all object definitions sheets (to a maximum of 2 marks)</p> <p>Default Properties with values</p> <ul style="list-style-type: none"> ○ 2 marks: Most or all default properties listed, and with correct values. ○ 1 mark: Some properties missing, or with incorrect values. ○ 0 marks: Not attempted or no properties listed correctly. 	<u>2 marks</u>
	<p>Event Procedures (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> ○ 2 marks: Most or all event procedures listed correctly. ○ 1 mark: Some event procedures listed correctly. ○ 0 marks: Not attempted or very few event procedures listed correctly. 	<u>2 marks</u>
2		<u>Total 20 marks</u>

Implementation

	Guide	Maximum Marks
3	<p>Build the Options form and write the code to make it functional (10 marks):</p> <ul style="list-style-type: none"> • Declare arrays to store the room types and costs (1 mark) • Declare arrays to store the food packages and costs (1 mark) • Declare variables to store the costs for the four optional packages (1 mark) • Create labels to identify the combo boxes for room type and food packages (1 mark) • Create combo boxes to store the room type options and food package options (1 mark) • Fill the room types array with the correct values (1 mark) • Fill the room prices array with the correct values (1 mark) • Fill the food packages array with the correct values (1 mark) • Fill the food prices array with the correct values (1 mark) • Assign the cost values for each of the optional packages in the correct variables (1 mark) 	<u>10 marks</u>
	<p>Use a for loop to loop over the room types array and use it to fill the room types combo box (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> • Declare the for loop and get it to repeat as many times as there are items in the array (1 mark) • Add the values stored at the current index of the room type and room cost arrays to the combo box (1 mark) 	<u>2 marks</u>
	<p>Use a for loop to loop over the food packages array and use it to fill the food packages combo box (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> • Declare the for loop and get it to repeat as many times as there are items in the array (1 mark) • Add the values stored at the current index of the room type and room cost arrays to the combo box (1 mark) 	<u>2 marks</u>
	<p>Write a function to calculate the total cost of the cruise based on the selected options (to a maximum of 11 marks):</p> <ul style="list-style-type: none"> • Declare a variable to store the total (1 mark) • Identify the selected index for the room types combo box (i.e. what is the index of the selected item) (1 mark) • Access the price in the room prices array at the selected index number and add it to the total (1 mark) • Identify the selected index for the food package types combo box (i.e. what is the index of the selected item) (1 mark) • Access the price in the food prices array at the selected index number and add it to the total (1 mark) • Check if the shore option has been selected and, if it has, add the shore cost to the total (1 mark) 	

	Guide	Maximum Marks
	<ul style="list-style-type: none"> • Check if the spa option has been selected and, if it has, add the shore cost to the total (1 mark) • Check if the fitness option has been selected and, if it has, add the shore cost to the total (1 mark) • Check if the internet option has been selected and, if it has, add the shore cost to the total (1 mark) • Enable the Purchase button (1 mark) • Write code to show the Customer Details and Receipt form (and hide the initial form) when the Purchase button is clicked (1 mark) 	<u>11 marks</u>
3		Total 25 marks
4	<p>Build the Details and Receipt form and write the code to make it functional (to a maximum of 6 marks):</p> <ul style="list-style-type: none"> • Create labels to identify the name, address, street, town/ city, postcode and credit card text boxes (max 2 marks – award 2 marks if all are created, 1 mark if some are added to the form, 0 marks if all are missing or not attempted) • Create text boxes to accept user input for the name, address, street, town/ city, postcode and credit card details (max 2 marks – award 2 marks if all are created, 1 mark if some are added to the form, 0 marks if all are missing or not attempted) • Create buttons to submit details and print receipt (1 mark) • Set the print receipt button to disabled (1 mark) 	<u>6 marks</u>
	<p>Write validation code for the name and street to ensure they are not empty (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> • Get the text from the appropriate textbox and check if it is empty (1 mark) • Display a message indicating the field must be completed if it is empty (1 mark) 	<u>2 marks</u>
	<p>Write validation code for the town/ city to ensure it is between 2 and 25 characters in length (to a maximum of 3 marks)</p> <ul style="list-style-type: none"> • Check whether the length of the text in the town/ city textbox is fewer than 2 characters in length (1 mark) • Check whether the length of the text in the town/ city textbox is greater than 25 characters in length (1 mark) • If the length is fewer than 2 or greater than 25 characters, display a message asking to re-enter (1 mark) 	<u>3 marks</u>
	<p>Write validation code for the postcode to ensure it is between 6 and 8 characters long (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> • Check whether the length of the text in the postcode textbox is between 6 and 8 characters in length (1 mark) 	

	Guide	Maximum Marks
	<ul style="list-style-type: none"> • If the length is not between 6 and 8 characters, display a message asking to re-enter (1 mark) 	2 marks
	<p>Write validation code for the credit card number to ensure it is 16 digits in length and is numeric (to a maximum of 2 marks)</p> <ul style="list-style-type: none"> • Check whether the length of the text in the credit card textbox is equal to 16 characters in length (1 mark) • Check if the value is an integer (1 mark) 	2 marks
	Store the customer details in variables once the inputs have been validated (1 mark)	1 mark
	Enable to Print Receipt button once all data has been successfully entered and validated (1 mark)	1 mark
	<p>Create the function to write the data to file (to a maximum of 8 marks)</p> <ul style="list-style-type: none"> • Import the System.IO package (1 mark) • Declare a StreamWriter object (1 mark) • Create a file with an appropriate name of type text (2 marks max – 1 mark for appropriate name, 1 mark for making it of type txt) • Get the required values from the previous form (1 mark) • Write the customer's details to the file (1 mark) • Write the cost of the cruise to the file (1 mark) • Close the connection to the file (1 mark) 	8 marks
4		Total 25 marks

Testing

	Guide	Maximum Marks
5a	<p>Design a testing strategy and complete a Test Log for each user input and program calculation:</p> <ul style="list-style-type: none"> • Normal tests • Extreme tests • Exceptional tests <ul style="list-style-type: none"> ○ 15 marks: Award full marks if there is a testing strategy for each user input option and calculation, which includes normal, extreme and exceptional test cases where appropriate for each item of functionality, documented in a fully completed Test Log. ○ 12 – 14 marks: Testing strategy is thorough and documented in a Test Log but one or two items of functionality have been omitted ○ 8 – 11 marks: Testing strategy and completed Test Log covers most items of functionality but does not include extreme or exceptional tests OR these are covered but some important functionality is not tested ○ 4 – 7 marks: Testing strategy covers less than half the functionality and only includes tests for normal data. Test Log may be missing some elements. ○ 1 – 3 marks: Nominal attempt at a testing strategy for at least one item of functionality. Test Log missing or very brief ○ 0 marks: No testing strategy for any of the procedures 	15 marks
5b	<p>Provide evidence, including sample screenshots, or tests (to a maximum of 5 marks)</p> <ul style="list-style-type: none"> • 3 marks: Comparison of actual outcomes with expected outcomes. (Deduct marks if no comparisons are made, or tests do not match what the program actually does Deduct one mark if they've made a good attempt at it but are missing a couple of important tests, deduct 2 marks if there are only one or two tests, deduct all 3 marks if nothing is correct or not attempted.) • 2 marks: Clear screen shots showing actual outcomes. (Deduct marks is screen shots show different outcomes from what is listed as test results, or if only a few screenshots are provided Deduct one mark if they've made a good attempt at it but more than a couple of tests are missing, deduct all 2 marks if nothing is correct or not attempted.) 	5 marks
Total available marks for testing		20

Learning Outcomes matrix

Task	Learning Outcomes assessed	Marker can differentiate between varying levels of achievement
Task 1	1, 2, 3, 4, 5	Yes
Task 2	1, 2, 3, 4, 5	Yes
Task 3	2, 3, 4, 5,	Yes
Task 4	2, 3, 4, 5, 6	Yes
Task 5	1	Yes

Grade descriptors

Learning Outcome	Pass	Merit	Distinction
Create project documentation	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Implement a program that uses data capture and validation	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Implement a program that uses sequential programming with different data types	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Implement a program that uses iteration and selection constructs	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Implement a program that uses file i/o	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Implement a program that uses arrays	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard