

# IB Computer Science Dossier Evaluation Form

Date Submitted: \_\_\_\_\_ Name of Author of Dossier: \_\_\_\_\_

Date Evaluated: \_\_\_\_\_ Name of Evaluator of Dossier: \_\_\_\_\_

Is there a **client programme** with a declaration (.h) file on Page# \_\_\_\_\_ and a separate implementation (.cpp) file on Page# \_\_\_\_\_? . . . .

Is there a **class** with a declaration (.h) file on Page# \_\_\_\_\_ and a separate implementation (.cpp) file on Page# \_\_\_\_\_? . . . . .

The **assigned dossier mark** out of 35 is calculated by multiplying a **Mastery Factor** *times* the **Assessment Evaluation**. Of eleven stipulated programming techniques, the mastery factor is a direct reflection of the number of those techniques appropriately applied in the dossier.

Number of Mastery Techniques in the Programme Dossier		
Mastery Factor	Standard Level Dossier	Higher Level Dossier
1.00	7	9, 10 or 11
0.75	6	7 or 8
0.50	4 or 5	5 or 6
0.25	0, 1, 2 or 3	0, 1, 2, 3 or 4

## IB Computer Science Techniques of Mastery

1. **Insert** a new data item into a **linked list or tree**. It is insufficient to add data only to the front or rear of a list. . . . . Page# \_\_\_\_\_

2. **Delete** a data item from a **linked list or tree** and properly dispose the allocated memory. It is insufficient to delete a data item from only the front or from only the rear of a list. . . . . Page# \_\_\_\_\_

3. **Search** for a data item in a **linked list or tree**. . . . . Page# \_\_\_\_\_

4. **Add a new record** to a direct (random) access file. . . . . Page# \_\_\_\_\_

5. **Delete a record** from a direct (random) access file. . . . . Page# \_\_\_\_\_□

6. **Search for a record** in a direct (random) access file. . . . . Page# \_\_\_\_\_□

7. **Merge** two sorted data structures. . . . . Page# \_\_\_\_\_□

8. **User-Defined functions**. . . . . Page# \_\_\_\_\_□

9. **Use arrays, records or pointers as parameters**. . . . . Page# \_\_\_\_\_□

10. **Recursion** . . . . . Page# \_\_\_\_\_□

11. Use more than one kind of **composite** data structure **OR** use one **hierarchical composite** data structure. . . . . Page# \_\_\_\_\_□

A composite data structure is made from other data types, such as a record. A hierarchical composite data structure is where at least one element is an array of records. Another example is a record which has one field defined on another record or array.

**Total Number of the 11 Techniques Required by Higher Level IB Computer Science Found.** . . . . . \_\_\_\_\_

**Master Factor Based On Number of Mastery Techniques found in This Dossier. Circle one of:**    0.25    0.50    0.75    1.00

# Assessment Criteria of Programme Dossier

Date: \_\_\_\_\_ Author: \_\_\_\_\_ Evaluator: \_\_\_\_\_

	AB	IB
	↓	↓
	Maximum Mark	
<b>Table of Contents</b> .....	/3	/0
<b>Preface: Preamble</b> .....	/2	/0
<b>Preface: Acknowledgements</b> .....	/2	/0
<b>Preface: Annotated Bibliography</b> .....	/2	/0
<b>Title Pages</b> .....	/2	/0
<b>MT Mastery Techniques</b> .....	/9	/0
<b>Subtotal</b> .....	/20	/0
<b>A. Analysing the Problem:</b> A thorough discussion and analysis of the problem which is being solved. ....	/10	/3
<b>B. Document the Design Process:</b> structure charts, pseudocode using PURE, flow charts, instance diagrams .....	/10	/6
<b>C. Use Appropriate Data Structures:</b> fully support the data storage requirements of the problem .....	/5	/3
<b>D. Use Efficient Algorithms:</b> Executes rapidly and requires minimal storage, simple programming code. ....	/3	/3
<b>E. Designing a Testing Strategy:</b> Testing should describe and use a wide range of valid and invalid data .....	/5	/3
<b>F. Annotated Hard Copy of the Test Output:</b> A representative sample of sample runs of test cases .....	/5	/3

<b>G. User-Friendly features:</b>	includes design issues, screen dumps, helpful menus, help instructions . . . . .	/4	/2
<b>H. Handling Errors:</b>	Detect and reject erroneous data input from the user and run-time errors caused by calculations and data-file errors. . . . .	/4	/2
<b>I. Implementing the Program:</b>	programme functions well and is closely related to the design . . . . .	/6	/3
<b>J. Good Programming Style:</b>	Source Code. Easily readable, small and clearly structured modules with abundant comments, meaningful identifier names, consistent indentation and syntax highlighting_____	/8	/2
<b>K. Evaluation of Solutions:</b>	a critical analysis; efficiency in general terms, effectiveness in relation to the original description of the problem, realistic improvements and extensions. . . . .	/10	/3
<b>L. User Documentation:</b>	clear and thorough written instructions for launching and using the programme . . . . .	/10	/2

**Total Possible Marks From Assessment Evaluation Before Applying The Mastery Factor** \_\_\_\_\_ /100 /35

The **assigned dossier mark** out of 35 is calculated by multiplying a **Mastery Factor** *times* the **Assessment Evaluation**. Of eleven stipulated programming techniques, the mastery factor is a direct reflection of the number of those techniques appropriately applied in the dossier.

**Assigned Dossier Mark = Mastery Factor [     ] X Assessment Evaluation [     ] =            /35**

## **COLLABORATIVE WORK IS FORBIDDEN.**

A teacher is expected to give “educational guidance”, but the work must be “that of the candidate alone.”