Dossier Criterion B2: Algorithms

You should choose algorithms that fully support the processes needed to achieve the objectives of the solution and provide sufficient support for the required data structures.

|  |  |
| --- | --- |
| **Achievement Levels** | **Descriptor** |
| 0 | The student has not reached a standard described by any of the descriptors given below. |
| 1 | The student has **outlined some** of the algorithms to be used in the solution. |
| 2 | The student has **described most** of the algorithms to be used, with details of parameters and return values. |
| 3 | The student has **discussed all** of the algorithms to be used, with details of parameters, return values, pre-conditions, and post-conditions. |
| 4 | The algorithms discussed are sufficiently **logical, detailed, and well documented** to be used to create the solution in Java. |

This section would typically be two to five pages in length. This can be a list or outline of all algorithms, presented as text, possibly in outline format. Standard algorithms (such as search or sort) can simply be named (with parameters), but non-standard algorithms must be described in more detail.

What’s a parameter? The data coming INTO a function/method.

What’s a return value? The data going OUT of a function/method.

What’s a pre-condition? A condition that is true BEFORE the method runs

What’s a post-condition? A condition that is true AFTER the method runs

//Achievement Level 4 Example:

//Parameters: an integer n, of which the factorial: n! is calculated

//Return Values: an integer, which is the factorial of n

//Pre-condition: n is non-negative

//Post-condition: n! is small enough to be represented by an int

public int factorial(int n)  
{

if (n == 0)

return 1;

if (n == 1)

return 1:

return n\*factorial(n-1);

}