



Advanced Java

Subject: In Object-oriented Programming which is Better as a Super Class – an Interface or an Abstract Class? And Why?

Grading Rubric Instructions: each row in the critical life skills rubric below can have a value indicated by the column heading for “Points per Row”. For columns less than “Professional” point values may be awarded on a sliding scale. So, e.g., “Life/Acceptable” points could be somewhere between 4 – 5. Total points possible for this exercise is 25.

Student Name: _____

Critical Thinking Skills Rubric	Evaluate pertinent information to reach an informed conclusion in personal, academic, and professional settings.			
	Not Acceptable	Introductory/Novice	Life/Acceptable	Professional
Points per Row	0 - 2	3	4	5
Identify Topic	Needs assistance to understand the instructions for this exercise and why it's important to professional development. <u>Example:</u> A student complains that they don't understand Interfaces and Abstract classes and therefore can't complete the exercise without the benefit of the instructor's teachings. The instructor must explain that independent research is a key skill and that the instructor will augment this research later.	Understands the instructions for this exercise but needs assistance appreciating how and why to apply critical thinking. <u>Example:</u> The student expects to do programming work in a programming class and cannot comprehend why they should have to participate in a debate, which is typically a task in a Communications course. The instructor must explain that professional programmers are expected to do independent research, make conclusions, offer recommendations and defend those to their peers and supervisors.	Understands the instructions for this exercise and how and why to apply critical thinking, but struggles to understand how to frame the effort within a commercial perspective. <u>Example:</u> The student is able to deal with the technical issues but struggles to consider related real world issues such as time and budget constraints.	Understands the instructions for this exercise and how and why to apply critical thinking, and is able to address real world constraints and considerations as a working professional. <u>Example:</u> The student suggests that “Interfaces” are the best choice for maximizing flexibility and minimizing the risks of concrete inheritance, but suggests that time and budget constraints may make this choice untenable due to the lack of code reuse.
Gather and Evaluate Information	Needs assistance to show evidence of selection or source evaluation skills	Demonstrates minimal ability to select and evaluate sources	Evaluates and questions relevance, accuracy, and bias of sources	Evaluates and questions relevance, accuracy, and bias of sources from a



				<p>professional Perspective</p> <p><u>Example:</u> The student explains support for using Interfaces based on several research sources and contrasts those with several other dissenting source, pointing out where arguments for/against are fair or inappropriate. For example, although Interfaces are well-documented to be slower than Abstract Classes, the speed difference is negligible in a real world application, while the flexibility and inheritance-safety benefits are tangible.</p>
Utilize Pertinent Information to Reach Conclusion	Needs assistance to use pertinent information	Uses some pertinent information without assistance	Uses ample pertinent information without assistance	<p>Uses professionally acknowledged pertinent information</p> <p><u>Example:</u> The student supports some conclusions with profit/loss calculations and compares those to loss of business cost due to customer dissatisfaction.</p>
Evaluate Results	Needs assistance to evaluate results	Attempts to evaluate results with minimal assistance	Evaluates results and considers the implications without assistance	<p>Evaluates results and considers the implications using professional standards</p> <p><u>Example:</u> The student suggests an overall winner</p>



				(Interface or Abstract Class) but frames this conclusion in reality, stating that the best choice is really a requirements issue and that programmers should use the best tool for the job. If you treat every job as requiring a hammer you will see every problem as a nail.
--	--	--	--	--