Mid Exam.pdf

To evaluate the mid-exam questions and solutions regarding their alignment with Bloom's Taxonomy levels and provide necessary feedback, let's analyze each question and its corresponding solution systematically.

Question Evaluation

Question 1:

1. **Question**: Create a Java class representing a Circle with a private radius attribute. Include public methods to set the radius and calculate the area of the circle. [CLO-1]

•**Identified Taxonomy Level**: Remembering and Understanding

•**Improvement Suggestions**: This question is low-level as it involves simple coding without deeper understanding or application. To elevate it:

•Reword it to include a requirement for error handling (e.g., handle negative radius).

•Example Questions:

1. Create a Java class representing a Circle with a private radius attribute. Include methods to set the radius validating that it is non-negative, calculate the area of the circle, and return a string representation of the circle.
2. In addition to the Circle class, create a method that compares the area of two Circle objects and determines which one is larger.

Question 2:

1. **Question**: Define a Java class Animal with a method makeSound(). Create a subclass Dog that overrides the makeSound() method to print "Dog barks". [CLO-1]

•**Identified Taxonomy Level**: Understanding and Applying

•**Comments**: This question is appropriate, but could further engage students by asking them to implement an additional subclass, allowing deeper exploration of polymorphism.

Question 3:

1. **Question**: Create an interface Shape with a method calculateArea(). Implement this interface in classes Circle and Rectangle. Test polymorphism by calling calculateArea() on instances of both classes. [CLO-1]

•**Identified Taxonomy Level**: Applying

•**Comments**: This question aligns with the required Bloom's level; no significant changes needed. However, additional context for practical applications of polymorphism could enhance complexity.

Question 4:

1. **Question**: What is encapsulation and why is it important in OOP? How does encapsulation promote data hiding? Give an example of encapsulation in Java with a class containing private variables and public methods. [CLO-1]

•**Identified Taxonomy Level**: Understanding and Analyzing

•**Comments**: This question meets the intended level. Encouraging students to review existing classes to identify encapsulation could enhance its application aspect.

Question 5:

1. **Question**: What is polymorphism and how does it contribute to code reusability? Explain the difference between compile-time polymorphism and runtime polymorphism. Give an example of polymorphism in Java using method overriding. [CLO-1]

•**Identified Taxonomy Level**: Analyzing and Evaluating
•**Comments**: This question appropriately challenges students and provides a solid basis for deeper analysis into polymorphism nuances.

Solution Evaluation

General Note on Solutions: The solutions provided generally reflect an understanding of core OOP concepts and give examples that demonstrate the requested knowledge. Here is a breakdown of individual solutions and their evaluations:

1. Solution for Circle class:
•**Rating**: 7/10
•**Key Elements**: Correctly defines the Circle class and includes necessary methods.
•**Suggestions for Improvement**: Add validation for radius.

2. Solution for Animal and Dog classes:
•**Rating**: 8/10
•**Key Elements**: Polymorphism demonstrated in implementation via subclassing.
•**Suggestions for Improvement**: Include a method to make sound of other Animal subclasses to further illustrate polymorphism.

3. Implementation of Shape interface:
•**Rating**: 9/10
•**Key Elements**: Correctly implements polymorphism through the Shape interface.
•**Suggestions for Improvement**: Add error handling in implementation to enhance robustness.

4. Explanation of Encapsulation:
•**Rating**: 8/10
•**Key Elements**: Clearly defines encapsulation and provides a valid example class.
•**Suggestions for Improvement**: Include a scenario where encapsulation effectively safeguards data.

5. Polymorphism Explanation:
•**Rating**: 9/10
•**Key Elements**: Thorough explanation of polymorphism and its effects on reusability.
•**Suggestions for Improvement**: Provide a more detailed comparison between compile-time and runtime with practical examples.

Conclusion

Overall, the exam is well-structured with clarity in questions. Small enhancements could push the evaluations and expectations further up Bloom's Taxonomy, particularly focusing on application and analysis tasks. Constructive suggestions were provided for both questions and answers to refine the assessments.
