Project 5 Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_

|  |  | | |
| --- | --- | --- | --- |
| **Item**  **Number** | **Criteria** | **Ratings** | **Score** |
| 1 | Mammal interface correctly created | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  The Mammal class is an interface that has at least 3 abstract methods | 5 pts  Partial Credit  The Mammal class is not an interface or it does not have at least 3 abstract methods | 0 pts  No Marks  The Mammal class is not an interface and it does not have at least 3 abstract methods | |  |
| 2 | There are at least 3 abstract classes extended from the Mammal class – Cat, Dog and Rabbit | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  There are 3 or more abstract classes extended from the Mammal class | 5 pts  Partial Credit  There are at least 2 abstract classes extended from the Mammal class from the required abstract classes | 0 pts  No Marks  There are 1 or no abstract classes extended from the Mammal class | |  |
| 3 | The concrete animal classes extend from an abstract class based on Mammal with one of the classes extended from a concrete class | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  The concrete animal classes extend from an abstract class based on Mammal with one of the classes extended from a concrete class | 5 pts  Partial Credit  The concrete animal classes extend from an abstract class based on Mammal or one of the classes extended from a concrete class | 0 pts  No Marks  The concrete animal classes are not extended from an abstract class based on Mammal | |  |
| 4 | The abstract classes or concrete classes implement the Comparable<T> interface and they have a compareTo() method | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  The abstract classes or concrete classes implement the Comparable<T> interface and they have a compareTo() method | 5 pts  Partial Credit  The abstract classes or concrete classes do not implement the Comparable<T> interface or they do not have a compareTo() method | 0 pts  No Marks  The abstract classes or concrete classes do not implement the Comparable<T> interface nor do they have a compareTo() method | |  |
| 5 | The subclasses use private instance variables correctly including appropriate constructor(s) and have at least 5 attributes | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  The subclasses use private instance variables correctly including appropriate constructor(s) and have at least 5 attributes | 5 pts  Partial Credit  The subclasses use private instance variables correctly including appropriate constructor(s) or the class has at least 5 attributes | 0 pts  No Marks  The subclasses do not use private instance variables correctly nor is there an appropriate constructor and there is not at least 5 attributes | |  |
| 6 | The tester class has a data structure that uses polymorphism to store objects | |  |  | | --- | --- | | 10 pts  Full Marks  The tester class has a data structure that uses polymorphism to store objects | 0 pts  No Marks  The tester class does not use a data structure that uses polymorphism to store objects | |  |
| 7 | The output includes a menu allowing the user to print the information stored in the instance variables in the super class and the subclasses, as well as the result of the comparison using the compareTo() method as appropriate | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  The output includes a menu allowing the user to print the information stored in the instance variables in the super class and the subclasses, as well as the result of the comparison using the compareTo() method as appropriate | 5 pts  Partial Credit  The output does not include a menu or it does not include the information stored in the instance variables in the super class and the subclasses nor does it include the result of the comparison using the compareTo() method | 0 pts  No Marks  The output does not include a menu nor does it include the information stored in the instance variables in the super class and the subclasses nor does it include the result of the comparison using the compareTo() method | |  |
| 8 | Code compiles and runs without errors | |  |  | | --- | --- | | 5 pts  Full Marks  Code compiles and runs without errors | 0 pts  No Marks  Code does not compile and is automatically assigned 0 points for the overall grade | |  |
| 9 | Code is properly commented | |  |  |  | | --- | --- | --- | | 10 pts  Full Marks  Code is properly commented including all required header information | 5 pts  Partial Credit  Code has some of the required comments but is sparsely commented or missing the correct header information | 0 pts  No Marks  Code is sparsely commented or not commented at all and is missing all or most of the correct header information | |  |
| 10 | Program is properly designed and uses whitespace effectively | |  |  |  | | --- | --- | --- | | 15 pts  Full Marks  Program is properly designed and uses whitespace effectively | 8 pts  Partial Credit  Program is not properly designed or it does not use whitespace effectively | 0 pts  No Marks  Program is not properly designed and it does not use whitespace effectively | |  |

Comments: