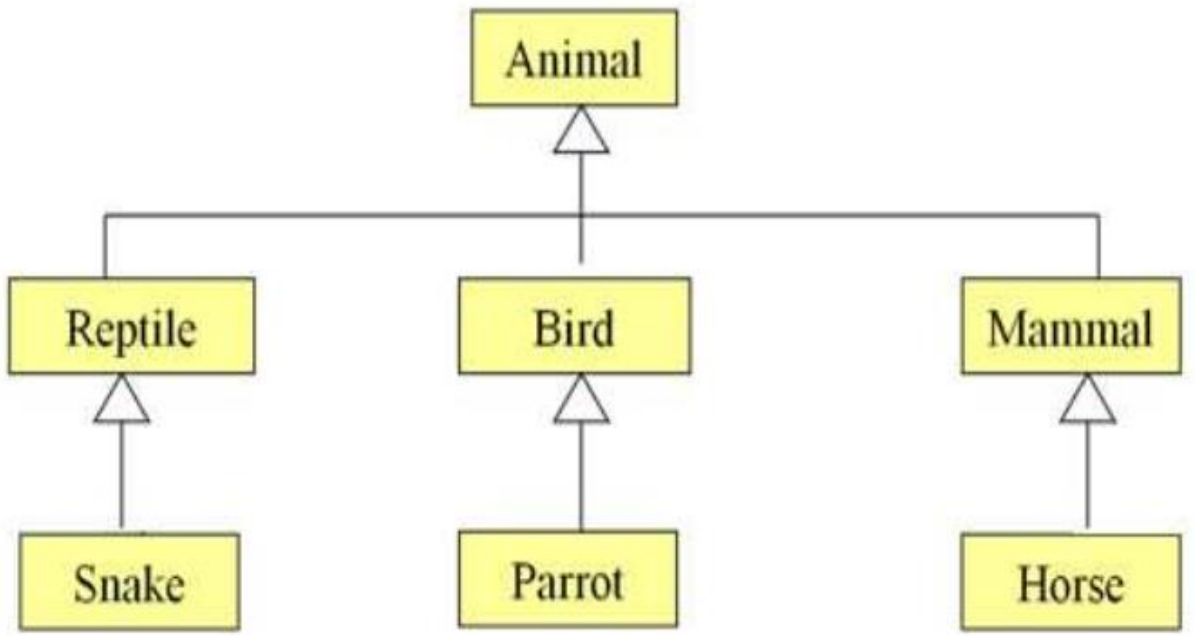
Task 1



Implement the above class hierarchy in Java with inheritance.

Task_1A:

Add a protected data member named "id" of type integer to the Animal class.

Task_1B:

Add public getter and setter methods for the "id" data member in the Animal class.

Task_1C:

Add a default constructor to each of the above-created classes. The constructor should initialize the "id" data member of Animal, Reptile, Bird, and Mammal to 0, 1, 2, and 3 respectively. It should also display the message "(class Name)'s default constructor."

Task_1D:

Add a parameterized constructor to each of the above-created classes. The constructor should initialize the "id" data member of Animal, Reptile, Bird, and Mammal to the specified id passed to the constructor. It should also display the message "(class Name)'s default constructor." The parameterized constructor should call its base class's parameterized constructor to initialize any data member inherited from the base class.

Task_1E:

Add a "getFamily" method to the Snake, Parrot, and Horse classes. This method must return the id of the particular animal's family it belongs to.

Task_1F:

Add a "tellAboutSelf" method to the Animal class with public access. The method should display the name of each object's class it belongs to.

Task_1G:

Override the "tellAboutSelf" method in all the subclasses of Animal with the same functionality as in the Animal class.

Task_1H:

Add a "speak" method to the Animal class and override it in each of the subclasses. The Animal, Reptile, Bird, Mammal, Snake, Parrot, and Horse can speak by printing the following messages: "I am an Animal, I can't speak...", "Family of Reptiles...", "Family of Birds...", "Family of Mammals...", "Shhh...", "Trrrr...", and "Hurrrrr..." respectively.

Task_1J:

Write your main method and instantiate an object of each class created above. Call the "tellAboutSelf", "speak", and "getFamily" methods for each created object.

Task 2

Instructions:

Work on this lab individually. You can use your books, notes, handouts, etc., but you are not allowed to borrow anything from your peers.

Marking Criteria:

Show your work to the instructor before leaving the lab to receive some or full credit.

What you have to do:

Create the following classes in Java using an Integrated Development Environment (IDE) of your choice. Compile and execute the program after completing each task. Name your files according to the task given in this lab.

Ship, CruiseShip, CargoShip, and BattleShip Classes:

1. Ship class:

Design a Ship class with the following members:

- A member variable for the name of the ship (a String).
- A member variable for the year that the ship was built (a String).
- A constructor(s) and appropriate getter and setter methods.
- A print() method that displays the ship's name and the year it was built.

2. CruiseShip class:

Design a CruiseShip class that extends the Ship class. The CruiseShip class should have the following members:

- A member variable for the maximum number of passengers (an int).
- A constructor(s) and appropriate getter and setter methods.

- An overridden print() method that displays the ship's name, year of construction, and the maximum number of passengers.

3. CargoShip class:

Design a CargoShip class that extends the Ship class. The CargoShip class should have the following members:

- A member variable for the cargo capacity in tonnage (an int).
- A constructor(s) and appropriate getter and setter methods.
- An overridden print() method that displays only the ship's name and the ship's cargo capacity.

4. BattleShip class:

Design a BattleShip class that extends the Ship class. The BattleShip class should have the following members:

- A member variable for the total number of missiles (an int).
- A constructor(s) and appropriate getter and setter methods.
- An overridden print() method that displays only the ship's name and the missile capacity.

5. A Driver Program:

Demonstrate the classes in a program that uses an array of Ship references. Initialize the array elements with dynamically allocated Ship, CruiseShip, CargoShip, and BattleShip objects. Then, iterate through the array, calling each object's print() method.