Single Inheritance

1) Create a parent class called "Vehicle" with a method "start_engine" that prints out "The engine is starting". Create a child class called "Car" that inherits from "Vehicle" and has an additional method "drive" that prints out "The car is driving".

2) Create a parent class called "Employee" with attributes "name" and "salary". Create a child class called "Manager" that inherits from "Employee" and has an additional attribute "department".

3) Create a parent class called "Shape" with attributes "color" and "size". Create a child class called "Circle" that inherits from "Shape" and has an additional attribute "radius".

4) Create a parent class called "Person" with attributes "name" and "age". Create a child class called "Student" that inherits from "Person" and has an additional attribute "grade".

5) Create a parent class called "Animal" with attributes "species" and "legs". Create a child class called "Cat" that inherits from "Animal" and has an additional attribute "color".

Multi Level Inheritance Questions

1) Create a class Person with a constructor that takes a name and age argument and stores them in instance variables. Create a subclass Employee that inherits from Person and has a constructor that takes a salary argument and stores it in an instance variable. Create a subclass Manager that inherits from Employee and has a constructor that takes a department argument and stores it in an instance variable.

2) Create a class Shape with a constructor that takes a color argument and stores it in an instance variable. Create a subclass Rectangle that inherits from Shape and has a constructor that takes width and height arguments and stores them in instance variables. Create a subclass Square that inherits from Rectangle and has a constructor that takes a side argument and sets the width and height instance variables to the value of side.

Multiple Inheritance Questions

1) Create a class Animal with a constructor that takes a name argument and stores it in an instance variable. Create a class CanFly with a method fly() that doesn't do anything. Create a class CanSwim with a method swim() that doesn't do anything. Create a subclass Duck that inherits from Animal, CanFly, and CanSwim and has a constructor that takes a color argument and stores it in an instance variable.

2) Create a class Person with a constructor that takes a name argument and stores it in an instance variable. Create a class CanCook with a method cook() that doesn't do anything. Create a class CanPaint with a method paint() that doesn't do anything. Create a subclass Chef that inherits from Person, CanCook, and CanPaint and has a constructor that takes a specialty argument and stores it in an instance variable.

Hierarchial Inheritance Questions

1) Create a class Vehicle with instance variables brand and model, and a method start_engine() that prints a message that the engine has started. Create a class Car that also inherits from Vehicle and has additional instance variables color and seats, and a method drive() that prints a message that the car is being driven. Create a class Bike that also inherits from Vehicle and has additional instance variables color and type, and a method ride() that prints a message that the bike is being ridden. Create objects of both Car and Bike classes and call their respective methods.

2) Create a class Device that has instance variables name and price, and a method turn_on() that prints a message indicating that the device is turning on. Create subclasses Phone and Tablet that inherit from Device and add a method make_call() to Phone and a method browse_internet() to Tablet. Create objects of the Phone and Tablet classes and call their turn_on(), make_call(), and browse_internet() methods.

3) Create a class Person that has instance variables name and age, and a method speak() that prints a message indicating that the person is speaking. Create subclasses Student and Teacher that inherit from Person and add a method study() to Student and a method teach() to Teacher. Create objects of the Student and Teacher classes and call their speak(), study(), and teach() methods.