**Single Inheritance Examples**

**Scenario-1**

**Employee and Manager:**

**Description:** Create a class Employee with attributes like name, id, salary, and methods like getDetails() and calculateSalary(). Then, create a subclass Manager that inherits from Employee and adds additional features like bonus and manageTeam().

**Real-world scenario:** A Manager is a type of Employee with extra responsibilities.

**Scenario-2**

**Animal and Dog:**

**Description:** Define a class Animal with basic attributes and methods like speak() and move(). Create a subclass Dog that inherits from Animal and adds methods like bark() and fetch().

**Real-world scenario:** A Dog is an Animal with additional behaviours.

Scenario-3

**Person and Student:**

**Description:** Create a class Person with attributes like name, age, and address, and methods like displayInfo(). Then, create a subclass Student that adds additional attributes like studentId, grade, and a method study().

**Real-world scenario:** A Student is a type of Person with more specific information.

Scenario-4

**Shape and Circle:**

**Description:** Define a class Shape with basic properties like area and perimeter and a method calculateArea(). Then, create a subclass Circle that inherits from Shape and includes specific details like radius and a method calculateCircumference().

**Real-world scenario:** A Circle is a specific type of Shape.

Scenario-5

**Vehicle and Car:**

**Description:** Define a class Vehicle with attributes like make, model, speed and methods like accelerate() and brake(). Then, create a subclass Car that adds specific methods like openSunroof() or playMusic().

**Real-world scenario:** A Car is a specialized type of Vehicle.

**Multilevel Inheritance Examples:**

**Scenario-6**

**Person, Student, and GraduateStudent:**

**Description:** Define a class Person with general information like name, age, and methods like displayInfo(). Then, create a Student class that inherits from Person, with additional attributes like studentId and methods like study(). Finally, create a GraduateStudent class that inherits from Student and adds more specific methods like conductResearch().

**Real-world scenario:** A GraduateStudent is a specialized type of Student, which in turn is a type of Person.

Scenario-7

**Animal, Bird, and Parrot:**

**Description:** Create a class Animal with basic attributes and methods like eat(), sleep(). Then, create a subclass Bird that inherits from Animal and adds methods like fly(). Finally, create a subclass Parrot that inherits from Bird and includes specific methods like talk() and mimic().

**Real-world scenario:** A Parrot is a type of Bird, and a Bird is a type of Animal.

Scenario-8

**Appliance, Refrigerator, and SmartRefrigerator:**

**Description:** Define a class Appliance with attributes like brand, powerRating, and methods like turnOn() and turnOff(). Create a subclass Refrigerator with additional attributes like temperatureControl and methods like setTemperature(). Then, create a subclass SmartRefrigerator with advanced features like connectToWiFi() and remoteControl().

**Real-world scenario:** A SmartRefrigerator is a type of Refrigerator, which in turn is a type of Appliance.

Scenario-9

**Person, Employee, and Manager:**

**Description:** Start with a class Person that includes general attributes like name, address, and a method displayInfo(). Create a subclass Employee that adds attributes like employeeId, salary, and methods like work(). Then, create a subclass Manager that inherits from Employee and adds responsibilities like manageTeam().

**Real-world scenario:** A Manager is an Employee, and an Employee is a Person.