1. Inheritance Pseudocode :-

```
PROCEDURE Inheritence
Animal
    name: string
    method makesound()
    Print "the animal makes a sound"
    end method
    end Animal
    Dog inherits from Animal
      override method makesound()
      Print "the dog barks"
      end method
      end class
    Main
      my dog : Dog // declare
      Call my dog. makesound.
      End Program.
```

Program :-

```
Class Animal {
    String name;
    Public void makesound () {
        System.out.Println ("Animal make sound");
    }
}
class Dog extends Animal {
    Public void make sound() {
        System.out.Println("Dog barks");
    }
}
Public class Main {
    Public Static void main (String[] args) {
```

```
Animal animal = new Animal();
Animal.name = "Generic Animal";
animal.makeSound();

        Dog dog = new Dog();
        dog.name = "Buddy";
        dog.makeSound();
    }
}
```

2) Constructor inheritance :-

Program :-

```
Class Person {
  String name;
  int age;
    Public Person(String name, int age) {
    this.name = name;
    this.age = age;
    }
    Public void displayInfo() {
      System.out.println("Name: " + name + ", Age: " + age);
  }
}
class Student extends Person {
    String grade;
    Public Student(String name, int age) {
      Super(name, age);
```

```
        this.grade = grade;
    }
    Public void displayInfo();
        System.out.Println("grade:"+grade);
    }
}
Public Class Main(){
        Public Static void main (String[]args) {
            Student Student = new Student("Alice:" 20; "A");
                Student.displayInfo()
    }
}
```

3) Multilevel inheritance :-

Program :-

```
Class vehicle {
    string brand;
    Public vechile (String brand) {
    this.brand = brand;
    }
    Public void displayInfo() {
        System.out.Println("Brand:" + brand);
    }
}
class Car extends vehicle {
        String Model;
        Public Car (String brand, string Model) {
        Super(brand);
        this model = model;
    }
```

```
Public void displayInfo() {
    super.displayInfo();
    System.out.println("Battery Capacity: " + batteryCapacity +
    " kWh");
  }
}
Public static void main (String[] args){
  ElectricCar car = new ElectricCar("Tesla", "Model", 100);
 }
}
```

4) **Method overriding in inheritance :-**

Program :-

```
Class Shape {
    Public void draw() {
      System.out.Println("Drawing a shape");
  }
}
class Circle Extends Shape {
    Public void draw() {
      System.out.println("Drawing a circle");
  }
}
Public class Main {
    Public static void main (String[] args) {
        Shape shape = new Shape();
        Shape.draw();
```

```
        Circle circle = new Circle();
        Circle.draw();
        Rectangle rectangle = new Rectangle();
        Rectangle.draw();
    }
}
```

5) inheritance and Access Modifiers :-

Program :-

```
class Employee {
    Private int id;
    Protected String name;
    Public double Salary;
    Public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.Salary = Salary;
    }
    Public void displayManagementInfo() {
        System.out.println("ID:" + id);
        System.out.println("Name:" + name);
        System.out.println("Salary:" + salary);
    }
    Public class Main {
        Public Static void main(String[] Jargs) {
            Manager Manager = new Manager(w1, "Alice", 75000.0);
            Manager.displayManagerInfo();
            Manager.displayEmployeeInfo();
        }
    }
```