1. Write a class `Student` with fields `name` and `age`. Create a constructor that uses the `this` keyword to distinguish between instance variables and parameters.

2. Create a class `Rectangle` with two constructors: one that accepts both width and height, and another that only accepts width and sets a default height. Use the `this()` constructor chaining to avoid redundancy.

3. Write a class `Chain` where methods `step1()`, `step2()`, and `step3()` return `this` to allow method chaining.

4. Create a parent class `Animal` with a method `makeSound()` and a subclass `Dog` that overrides this method. Use the `super` keyword to call the parent class's method in the overridden version.

5. Write a class `Person` with a constructor that accepts `name` and `age`. Create a subclass `Employee` that accepts `name`, `age`, and `salary`, and use `super()` to initialize the name and age.

6. In the class `Vehicle` with a method `move()`, create a subclass `Car` that overrides `move()`. Use `super.move()` to call the superclass version inside the overridden method.

7. Write a class `Parent` with a method `display()`, and a subclass `Child` with a constructor that uses `super()` to call the parent class constructor. Use both `super.display()` and `this.display()` in the child class.

8. Create a class `Building` with overloaded constructors. Then create a class `House` that extends `Building`, and use `super()` to call different constructors from the superclass based on input parameters.

9. Create a class with an inner class and use `this` to refer to the outer class’s instance variables or methods.

10. Implement a `Human` class with two constructors. Create a `Student` class that calls both the `Human` constructors in different ways using `super()`.