Question 1, 2 and 3 are connected

1. **(Class Definition And Object Creation)**
   Create a simple Car class. This class should have three attributes: model, color, and year. Define a constructor that initializes these attributes and create a Car object.
2. **(Method Definition And Class Behavior)**
   Add a start_engine() method to the Car class you created above. When this method is called, it should print "Engine started".
3. **(Attribute Modification Through Methods)**
   Add a paint_car(new_color) method to the Car class. This method should update the color attribute of the car to the new_color provided and return the new color.
4. **(Memory Layout)**
   Draw the following code's memory layout drawing.

```java
class Person {
    String name;
    int age;

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public static void main(String[] args) {
        Person person = new Person("Emily", 30);
    }
}
```

5. **(Private Attributes And Getter/setter Methods)**
   Define a private attribute mileage in the Car class. Provide two methods get_mileage() and set_mileage(mileage) to access and update this attribute.
6. **(Memory Layout)**
   Draw the following code's memory layout drawing.

```java
class Engine {
    int horsepower;
}

class Car {
    String model;
    Engine engine;

    Car(String model, int horsepower) {
        this.model = model;
        this.engine = new Engine();
        this.engine.horsepower = horsepower;
    }
}

public class MemoryLayoutExample {
    public static void main(String[] args) {
        Car car1 = new Car("Sedan", 150);
        Car car2 = new Car("Coupe", 200);

        car1.engine.horsepower = 180;
    }
}
```

Question 7 and 8 are connected

7. **(Class With Attributes And Method Implementation)**
   Create a Circle class. This class should have a radius attribute and a calculate_area() method. The method should calculate and return the area of the circle.

8. **(Static Methods And Class Constants)**
   Add a "pi number" constant and a static method calculate_circumference(radius) to the Circle class. The method should calculate and return the circumference of the circle.
9. **[Method Implementation For Attribute Modification)**
   Create a Person class with name and age attributes. Add a have_birthday() method that increments the age by one when called.
10. **(Attributes)**
    Create a Student class with a grades list attribute. Write a method that updates this list and calculates the student's average grade.

Question 11 and 12 are connected

11. **(For Loop Question)**
    Write a class with a method that prints the following output. You will create a class named NumberPattern with a method called printPattern. In the main class, create an object of NumberPattern and then call the printPattern method.

    1

    1 2

    1 2 3

    1 2 3 4

    1 2 3 4 5

12. **(While Loop Question)**
    Add a method named evenSum to the previous code, which is inside the NumberPattern class. This method should calculate the sum of even numbers from 1 to 10 using a while loop. In the main class, use the previously created object to call the evenSum method.

Question 13,14 are connected

13. **(Constructors, attributes, class methods)**
    Create a class named "Book". Add three attributes: title, author, and pages.
    Write a constructor that initializes these attributes. Add a method getBookInfo() to the Book class. The method should return a string with the title, author, and number of pages.

14. **(if-else, main function, object creation, calling class methods)**
    For the Question 14 Book class, add a method isLongBook() The method should return true if the number of pages is greater than 300, otherwise false. Create a class called “Test”, and implement a main function to execute your code. Create two book objects in the main function. Call getBookInfo() and isLongBook() functions for these objects.

Question 15,16,17, 18 are connected

15. **(Constructors, attributes, class methods, setters)**
    Create a class named Student. Add attributes: name (String), grade (int), and attendance (int). Write a constructor that initializes these attributes. Add a method updateGrade(int newGrade) in the Student class. This method should update the student's grade.

16. **(function overloading, if-else)**
    Overload the updateGrade() method in the Student class in Question 16. Create another version that takes no parameters and increases the grade by 5. If the grade is more than 95, then the grade is set to 100.

17. **(if-else, access modifiers, getters-setters)**
    Create a method checkAttendance() in the Student class in question 16. If attendance is below 75%, return the string "Needs Improvement", otherwise return the string "Good". Attribute “grade” is now private. Implement the necessary functions to access and update the grade attribute.
18. **(arrays, for loops, attributes)**
    In the Student class of Question 16, add a method addAttendance(int[] days). This method should increase the attendance by the total of the days array.
19. **(function overloading, classes, class functions)**
    Create a Calculator class with overloaded methods add(). One method should take two integers, and another should take three integers.
20. **(default constructors, multiple constructors, class functions)**
    Create a Rectangle class with a constructor that takes width and height as parameters. Add methods getArea() and getPerimeter() to calculate and return the area and perimeter of the rectangle. Create another constructor for the Rectangle class with no parameters. Set default values of width = 1 and height = 1.
21. **(classes, for loops, strings)**
    Create a class StringUtilities with a method reverseString(String input). This method should return the reverse of the given string using a loop.
22. **(recursion, classes)**
    Create a class FactorialCalculator with a method calculate(int n). The method should use recursion to calculate and return the factorial of a given integer n. Ensure the method handles the base case where n is 0 or 1.