Chapter 03 Assignments (Easy difficulty)

Here are some easy Java 17 programming assignments that focus on introduction of OOP concept:

1. Create a class called "Rectangle" with attributes "width" and "height". Add a constructor that takes both "width" and "height" as parameters and initializes the attributes. Implement a method called "getArea" that returns the area of the rectangle. Implement a second constructor that takes only one parameter, "side", and creates a square with that side length by calling the first constructor.

2. Create a program that has a "Cube" class which calculates the surface area and volume of a cube given its side length.

3. Write an OOP program that computes the surface area and volume of a sphere given its radius. (Please solve this question by OOP concept)

4. Create a class called "Person" with attributes "name" and "age". Implement a method called "introduce" that prints "Hello, my name is {name} and I am {age} years old." Implement a constructor that takes both "name" and "age" as parameters and initializes the attributes.

5. Create a class called "Circle" with attributes "radius" and "color". Implement a method called "getArea" that returns the area of the circle. Implement a second method called "getCircumference" that returns the circumference of the circle. Implement a constructor that takes both "radius" and "color" as parameters and initializes the attributes.

6. Create a class called "Car" with attributes "make", "model", "year", and "speed". Implement a method called "accelerate" that increases the car's speed by a given amount. Implement a method called "brake" that decreases the car's speed by a given amount. Implement a method called "getCurrentSpeed" that returns the car's current speed. Implement a constructor that takes "make", "model", and "year" as parameters and initializes the attributes.

7. Create a class called "BankAccount" with attributes "balance" and "accountNumber". Implement a method called "deposit" that adds a given amount to the balance. Implement a method called "withdraw" that subtracts a given amount from the balance. Implement a method called "getBalance" that returns the current balance. Implement a constructor that takes "accountNumber" as a parameter and initializes the attribute.

8. Create a class called "TemperatureConverter" with static methods called "toFahrenheit" and "toCelsius". The "toFahrenheit" method takes a temperature in Celsius as a parameter and returns the temperature in Fahrenheit. The "toCelsius" method takes a temperature in Fahrenheit as a parameter and returns the temperature in Celsius.

9. Create a class called "MathUtils" with a static method called "factorial". The "factorial" method takes an integer as a parameter and returns the factorial of that integer.

10. Create a class called "StringUtils" with a static method called "reverse". The "reverse" method takes a string as a parameter and returns the reverse of that string.

11. Create a class called "Student" with attributes "name" and "grade". Implement a method called "display" that prints "Name: {name}, Grade: {grade}". Implement a constructor that takes "name" and

"grade" as parameters and initializes the attributes.

12. Create a class called "MathConstants" with final static variables called "PI" and "E". Implement a method called "calculateAreaOfCircle" that takes a radius as a parameter and returns the area of a circle with that radius using the "PI" constant. Extend your class with another method called "calculateRadiusOfCircle" that takes a double parameter as the circumference of the circle and returns the radius of the circle.