**Assignment on Classes and Objects**

1. Create a **Person** class with the following properties:

**String name**

**int age**

**String occupation**

1. Generate getters and setters for all properties of the **Person** class using the **@Field** annotation.
2. Add a **toString()** method to the **Person** class that returns a string representation of the object in the following format:

Person(name=<name>, age=<age>, occupation=<occupation>)

Sure, here's an assignment on Groovy classes and objects:

1. Create a **Person** class with the following properties:

* **String name**
* **int age**
* **String occupation**

1. Generate getters and setters for all properties of the **Person** class using the **@Field** annotation.
2. Add a **toString()** method to the **Person** class that returns a string representation of the object in the following format:

phpCopy code

Person(name=<name>, age=<age>, occupation=<occupation>)

1. Create a **PersonService** class that has the following methods:

* **Person createPerson(String name, int age, String occupation)** - Creates a new **Person** object with the specified properties and returns it.
* **void printPerson(Person person)** - Prints the string representation of a **Person** object to the console.

1. In the **createPerson()** method of the **PersonService** class, create a new **Person** object using the provided parameters, and return it.
2. In the **printPerson()** method of the **PersonService** class, use the **toString()** method of the **Person** class to print the string representation of the **Person** object to the console.
3. In the **Main** class, create a **PersonService** object and use it to create a new **Person** object with the following properties:

* **name** = "John Doe"
* **age** = 30
* **occupation** = "Software Developer"

1. Print the string representation of the **Person** object to the console using the **printPerson()** method of the **PersonService** class.
2. Create a new **Person** object with different properties and print its string representation to the console using the **printPerson()** method.
3. Bonus: Add validation to the **createPerson()** method to ensure that the **name** property is not empty and the **age** property is greater than zero. If either of these conditions is not met, throw an exception with a helpful error message.

This assignment will help you practice creating and working with classes and objects in Groovy. By creating a **Person** class and a **PersonService** class, we can demonstrate how to use objects and methods to manipulate and query data. The bonus task of adding validation to the **createPerson()** method allows students to practice error handling and input validation, which are important skills in any programming language