**Solution - Assignment on POGO**

import groovy.transform.Field

class Person {

@Field String name

@Field int age

@Field String occupation

String toString() {

"Person(name=${name}, age=${age}, occupation=${occupation})"

}

}

class PersonService {

Person createPerson(String name, int age, String occupation) {

if (name.isEmpty() || age <= 0) {

throw new IllegalArgumentException("Invalid name or age")

}

return new Person(name: name, age: age, occupation: occupation)

}

void printPerson(Person person) {

println person.toString()

}

}

class Main {

static void main(String[] args) {

PersonService service = new PersonService()

Person person1 = service.createPerson("John Doe", 30, "Software Developer")

service.printPerson(person1)

Person person2 = service.createPerson("Jane Smith", 25, "Data Analyst")

service.printPerson(person2)

}

}

Output

Person(name=John Doe, age=30, occupation=Software Developer)

Person(name=Jane Smith, age=25, occupation=Data Analyst)

The solution includes a Person class with name, age, and occupation properties generated using the **@Field** annotation. It also includes a **toString()** method that returns a string representation of the Person object. The PersonService class has **createPerson()** and **printPerson()** methods that create a new Person object using the provided parameters and print its string representation to the console, respectively. The Main class creates a PersonService object and uses it to create and print two Person objects with different properties. Additionally, the bonus task of adding validation to the **createPerson()** method to ensure that the name property is not empty and the age property is greater than zero is also implemented.