The provided code showcases constructor injection in the Spring Framework. It demonstrates the configuration and usage of beans using XML-based configuration.

The `**Test**` class serves as the main class, responsible for running the Spring application. It starts by creating an instance of `ClassPathXmlApplicationContext` using the XML configuration file's location. It then retrieves the bean with the ID "person" from the application context and assigns it to the variable `p`. The `p` object is printed using its `toString()` method, which internally invokes the `toString()` method of the `Person` class. Additionally, the bean with the ID "add" is retrieved from the application context, although no explicit actions are performed with it.

The `**Person**` class represents a person and contains properties such as `personName`, `personId`, `cert`, and `country`. It features a constructor that accepts parameters to initialize the `Person` object, including a `personName` (String), `personId` (int), `cert` (Certificate), and `country` (List of Strings). The `toString()` method is overridden to provide a string representation of the `Person` object.

The `**Certificate**` class represents a certificate and includes properties like `certificateName` and `certificateDuration`. It provides a constructor that takes parameters to initialize the `Certificate` object, including a `certificateName` (String) and `certificateDuration` (String). The `toString()` method is overridden to offer a string representation of the `Certificate` object.

The `**Addition**` class demonstrates a simple addition operation and contains properties `a` and `b`. It features three constructors that are overloaded to handle different types of input parameters: `double`, `int`, and `String`. Each constructor sets the values of `a` and `b` based on the provided parameters and prints a message indicating the constructor used. The `Addition` class is used to exemplify the ambiguity problem that can arise with constructor injection.

The `**ciconfig.xml**` file is an XML configuration file that defines the bean definitions and their dependencies. It begins by declaring the XML namespaces for Spring beans, context, p (property), and c (constructor-arg). The `<bean>` tag is utilized to define beans with their respective class and name attributes. The first bean definition is for the `Certificate` class, identified by the name "cer". The `c:certificateName` and `c:certificateDuration` attributes are employed to set the constructor arguments for the `Certificate` bean. The second bean definition is for the `Person` class, with the name "person". The `<constructor-arg>` tags set the constructor arguments for the `Person` bean. Some values are provided directly as literals, while the `cert` property is assigned the "cer" bean via a reference. The `country` property is set using a `<list>` of `<value>` tags. The third bean definition is for the `Addition` class, labeled as "add". The `<constructor-arg>` tags specify the constructor arguments for the `Addition` bean.

Regarding the **ambiguity problem**, it arises in the `**Addition**` class due to having multiple constructors with different parameter types. During the instantiation of the `Addition` bean, the Spring container must determine which constructor to use based on the provided constructor arguments. In the given code, the XML configuration file sets the constructor arguments as integers for the `Addition` bean. However, since there are multiple constructors that can accept integers, Spring may encounter ambiguity and throw an exception requesting clarification on the desired constructor.

To resolve this ambiguity problem, you can modify the XML configuration file by explicitly specifying the types of the constructor arguments using the `type` attribute. By indicating the parameter types, such as

`type="int"`, Spring can accurately determine the appropriate constructor to use based on the provided types. This resolves the ambiguity and ensures the correct constructor is invoked.

Person class:

```
package com.springcore.constructorinjection;
import java.util.List;
public class Person {
	private String personName;
	private int personId;
	private Certificate cert;
	private List<String> country;
	public Person(String personName, int personId, Certificate cert, List<String> country) {
		super();
		this.personName = personName;
		this.personId = personId;
		this.cert = cert;
		this.country = country;
	}
	@Override
	public String toString() {
		return "Person [personName=" + personName + ", personId=" + personId + ", cert=" + cert + ", country=" + country
				+ "]";
	}
}
```

Certificate class:

```
package com.springcore.constructorinjection;
public class Certificate {
	String certificateName;
	String certificateDuration;

	public Certificate(String certificateName, String certificateDuration) {
		super();
		this.certificateName = certificateName;
		this.certificateDuration = certificateDuration;
	}
	@Override
	public String toString() {
		return "Certificate [certificateName=" + certificateName + ", certificateDuration=" + certificateDuration + "]";
	}
}
```

Addition class:

```
package com.springcore.constructorinjection;
public class Addition {
		private int a;
		private int b;
		public Addition(double a, double b) {
			this.a = (int) a;
			this.b = (int) b;
			System.out.println("Constructor created for Double");
			System.out.println("a : "+a+" b : "+b);
		}
		public Addition(int a, int b) {
			this.a = a;
			this.b = b;
			System.out.println("Constructor created for Integer");
			System.out.println("a : "+a+" b : "+b);
		}


		public Addition(String a, String b) {
			this.a =  Integer.parseInt(a);
			this.b =  Integer.parseInt(b);
			System.out.println("Constructor created for String");
			System.out.println("a : "+a+" b : "+b);
		}

}
```

Main class:

```
package com.springcore.constructorinjection;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Test {
		public static void main(String[] args) {
			// TODO Auto-generated method stub
			ApplicationContext context = new ClassPathXmlApplicationContext("com/springcore/constructorinjection/ciconfig.xml");
			Person p = (Person) context.getBean("person");
			System.out.println(p);

			Addition addition = (Addition) context.getBean("add");
		}
}
```

Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
		xmlns:context="http://www.springframework.org/schema/context"
		xmlns:p="http://www.springframework.org/schema/p"
		xmlns:c="http://www.springframework.org/schema/c"
		xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
		xsi:schemaLocation="http://www.springframework.org/schema/beans
		https://www.springframework.org/schema/beans/spring-beans.xsd">
 <bean class="com.springcore.constructorinjection.Certificate" name="cer" c:certificateName="certificate" c:certificateDuration="duration"/>
 <bean class="com.springcore.constructorinjection.Person" name="person">
		<constructor-arg value="Ram"/>
		<constructor-arg value="12" type="int"/>
		<constructor-arg ref="cer"/>
		<constructor-arg>
			<list>
			<value>INDIA</value>
			<value>AFRICA</value>
		</list>
		</constructor-arg>
 </bean>
 <bean class="com.springcore.constructorinjection.Addition" name="add">
		<constructor-arg value="10" type="int" index="1"></constructor-arg>
		<constructor-arg value="20" type="int" index="0"></constructor-arg>
 </bean>
</beans>
```