

Home Assignment <2A>: Demonstrating Method Overloading in Java

Learning Objective:

To understand and implement the concept of **Overloading in Java** by creating a class and implementing methods with different sets of input arguments.

Expected Completion Time:

Best Case: 20 minutes

Average Case: 25 minutes

Assignment Details:

Create a Java class named **APIClient** and create two methods with the same name passing different input arguments.

Requirements:

- Inside the **APIClient** class, define the **sendRequest** method with multiple overloaded versions.
- One version should accept one input argument: a String for the **endpoint**.
- Another version of the **sendRequest** method should accept three input arguments: a String for the **endpoint**, a String for the **requestBody**, and a boolean parameter **requestStatus** to verify whether the request is successful.
- Create a main method to demonstrate the usage of the overloaded **sendRequest** method.
- Inside the main method, create an object of the **APIClient** class.
- Call both versions of the **sendRequest** method on the **APIClient** object with different sets of input arguments to showcase method overloading.

Hints to Solve:

- Include print statements inside each methods and pass input values
- Initialize objects using the new keyword and test the functionality by calling methods.

Reference Links:

Overloading in Java - [Oracle Docs](#)

Expected Outcome:

Upon completion, you should be able to:

- Grasp the fundamentals of method Overloading in Java.
- Understand the concept of compile time polymorphism

Home Assignment <2B>: Demonstrating Method Overriding in Java

Learning Objective:

To understand and implement the concept of **overriding in Java** by creating a generic class and a specific subclass.

Assignment Details:

Create a superclass with common methods for interacting with web elements. Implement a method from the superclass to provide a specific implementation in the subclass that overrides the superclass method.

Requirements:

- Create a Java class named **BasePage**
- Create methods like **findElement()**, **clickElement()**, **enterText()** and **performCommonTasks()**.
- Create a subclass named **LoginPage**.
- Override the **performCommonTasks()** method in the **LoginPage** class.
- Demonstrate the concept by creating objects for both classes and calling their methods.

Hints to Solve:

- Use the '**extends**' keyword for inheritance.
- The overridden method is annotated by **@Override**
- Initialize objects using the new keyword and test the functionality by calling methods.

Reference Links:

Method Overriding in Java - [Oracle Docs](#)

Expected Outcome:

Upon completion, you should be able to:

- Grasp the fundamentals of inheritance in Java.
- Create a subclass that inherits attributes and methods from a superclass.
- Override methods in a subclass.