



```
/*
The following method builds a home page based off reccomended sites
*/
public WebPage buildLandingPage(PersonalizedReccomendationList userList){
    WebPage wb = new WebPage();
    ReccomendationList list;
    User user = LocalStorage.getInstance().getLoggedInUser();
    if(user != null && userList.getUser.equals(user)){
        list = userList;
    }
    else{
        list = Database.getInstance().fetchDefaultReccomendations();
    }
    wb.add(list);
    return wb;
}

public TestPersonalizedReccomendationList extends
PersonalizedReccomendationList{
    public TestUser extends User{
        public bool equals(Object other){
            return true;
        }
    }
}

public User getUser(){
    return new TestUser();
}
}
```

Resolving Onion and Alias Parameters

- Non-necessary Parameters :
 - Pass `null` into the parameter field.
- Complex Parameters (Extract Interface Method):
 - Create an interface for the public methods of the troublesome parameter.
 - Make the original class implement the interface.
 - Fork a new class that also implements that interface and override necessary methods with more test-friendly variants.
 - Update the method to take the
- Deeply Nested Parameters (Subclass and Override Method):
 - Create a subclass of the class with the troublesome parameter.
 - Override the method you want to prevent aliasing.

```
/*
The following method builds a home page based off recommended sites
*/
public WebPage buildLandingPage(PersonalizedReccomendationList userList){
    WebPage wb = new WebPage();
    RecommendationList list;
    User user = LocalStorage.getInstance().getLoggedInUser();
    if(user != null && userList.getUser.equals(user)){
        list = userList;
    }
    else{
        list = Database.getInstance().fetchDefaultReccomendations();
    }
    wb.add(list);
    return wb;
}

public TestPersonalizedReccomendationList extends
PersonalizedReccomendationList{
    public TestUser extends User{
        public bool equals(Object other){
            return true;
        }
    }
}

public User getUser(){
    return new TestUser();
}
}
```

Demo Part 2