Zunera Zahid

lab-5

Abdulrab 369946 bese 12-a

Contents

[Assignment 2 1](#_Toc127962498)

[Bad Code Smell: Duplicate Code 2](#_Toc127962499)

[Example 2](#_Toc127962500)

[Solution: Extract methods, Deduplicate code, functions 2](#_Toc127962501)

[Solution: loops 2](#_Toc127962502)

[Bad Code Smell: Long method 2](#_Toc127962503)

[Example 2](#_Toc127962504)

[Solution: Extract Method, 2](#_Toc127962505)

[Solution: Replace Temp with Query, 2](#_Toc127962506)

[Solution: Introduce Parameter Object or Preserve Whole Object. 2](#_Toc127962507)

[Solution: Replace Method with Method Object. 2](#_Toc127962508)

[Solution: Decompose Conditional 2](#_Toc127962509)

[Bad Code Smell: Large class 2](#_Toc127962510)

[Example 2](#_Toc127962511)

[Solution: Extract Class, 2](#_Toc127962512)

[Solution: Extract Subclass, 2](#_Toc127962513)

[Solution: Extract Interface 2](#_Toc127962514)

[Solution: Duplicate Observed Data 2](#_Toc127962515)

## Assignment 2

Consider the following Bad code smells:

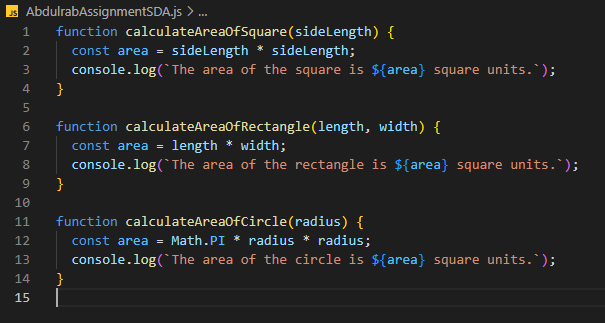
1. Duplicate Code
   * Solution: Extract methods, Deduplicate code, functions, loops
2. Long method
   * Solution Extract Method, Replace Temp with Query, Introduce Parameter Object or Preserve Whole Object. Replace Method with Method Object. Decompose Conditional
3. Large class
   * Solution: Extract Class, Extract Subclass, Extract Interface ,Duplicate Observed Data

Give example for each of these along with solution for each of the examples. Ensure adding a cover page with Name Section and Registration number. Also add a table of content including following headings:

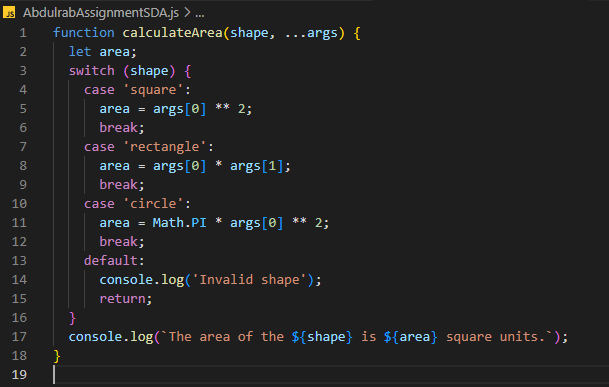
## Bad Code Smell: Duplicate Code

### Example

In the example that I chose all the functions are performing calculations based on geometry of different shapes. There is code duplication that can be reduced if we made one general function that can deal all of these calculations.

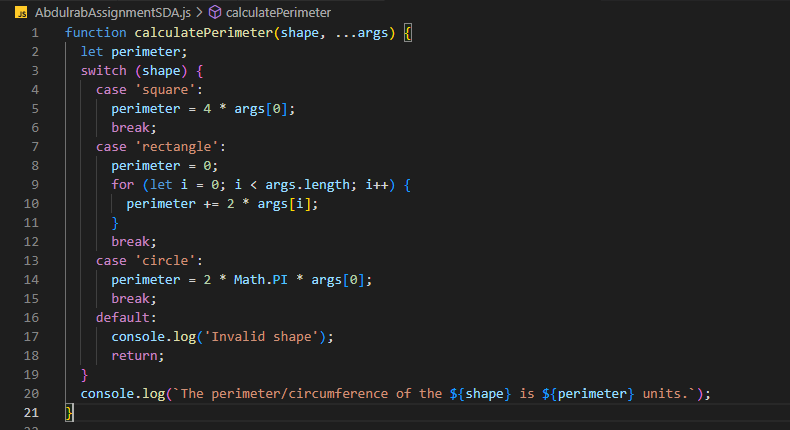


### Solution: Extract methods, Deduplicate code, functions



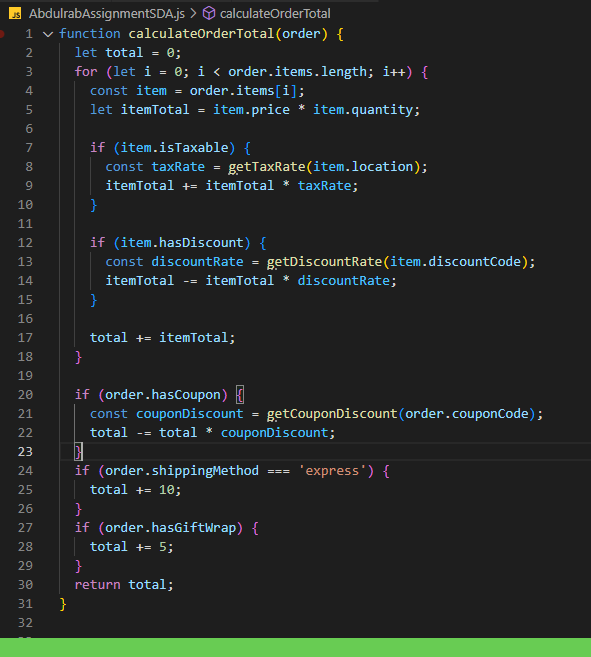
### Solution: loops

If we consider the case in which we calculate the perimeter of the above shapes, we can reduce duplication while integrating loops in our code:



## Bad Code Smell: Long method

### Example



### Solution: Extract Method,

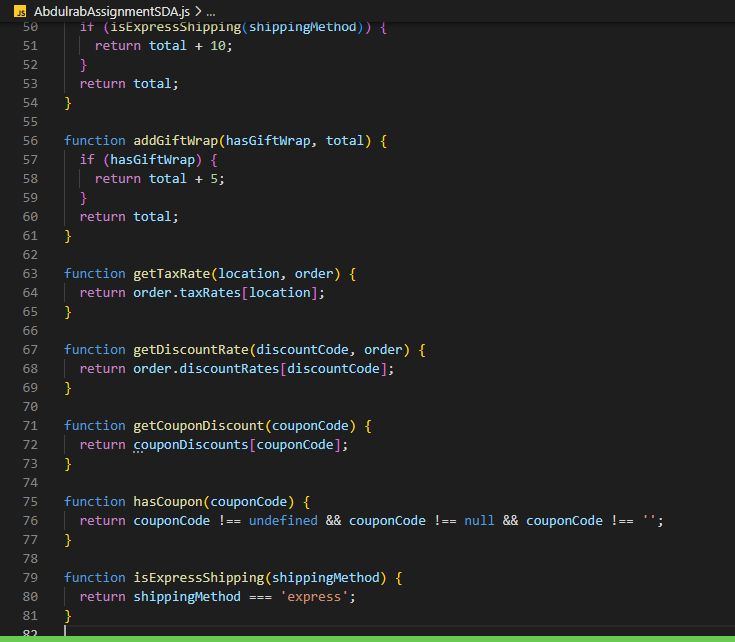




By breaking the code down into smaller, more manageable pieces, we can improve readability and make it easier to maintain and extend the code.

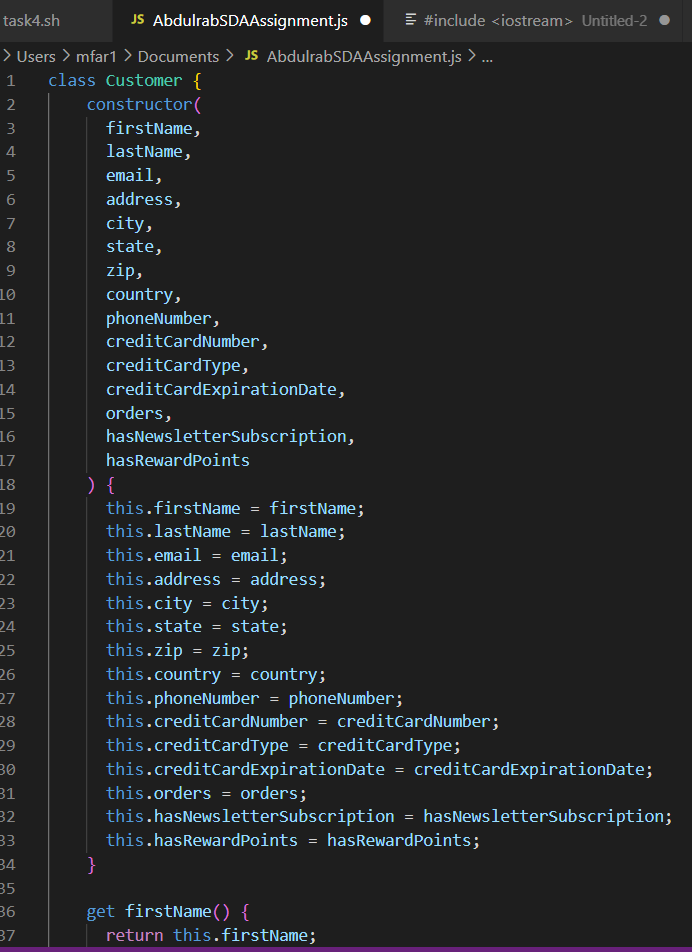
### Solution: Replace Temp with Query,





## Bad Code Smell: Large class

### Example



Text

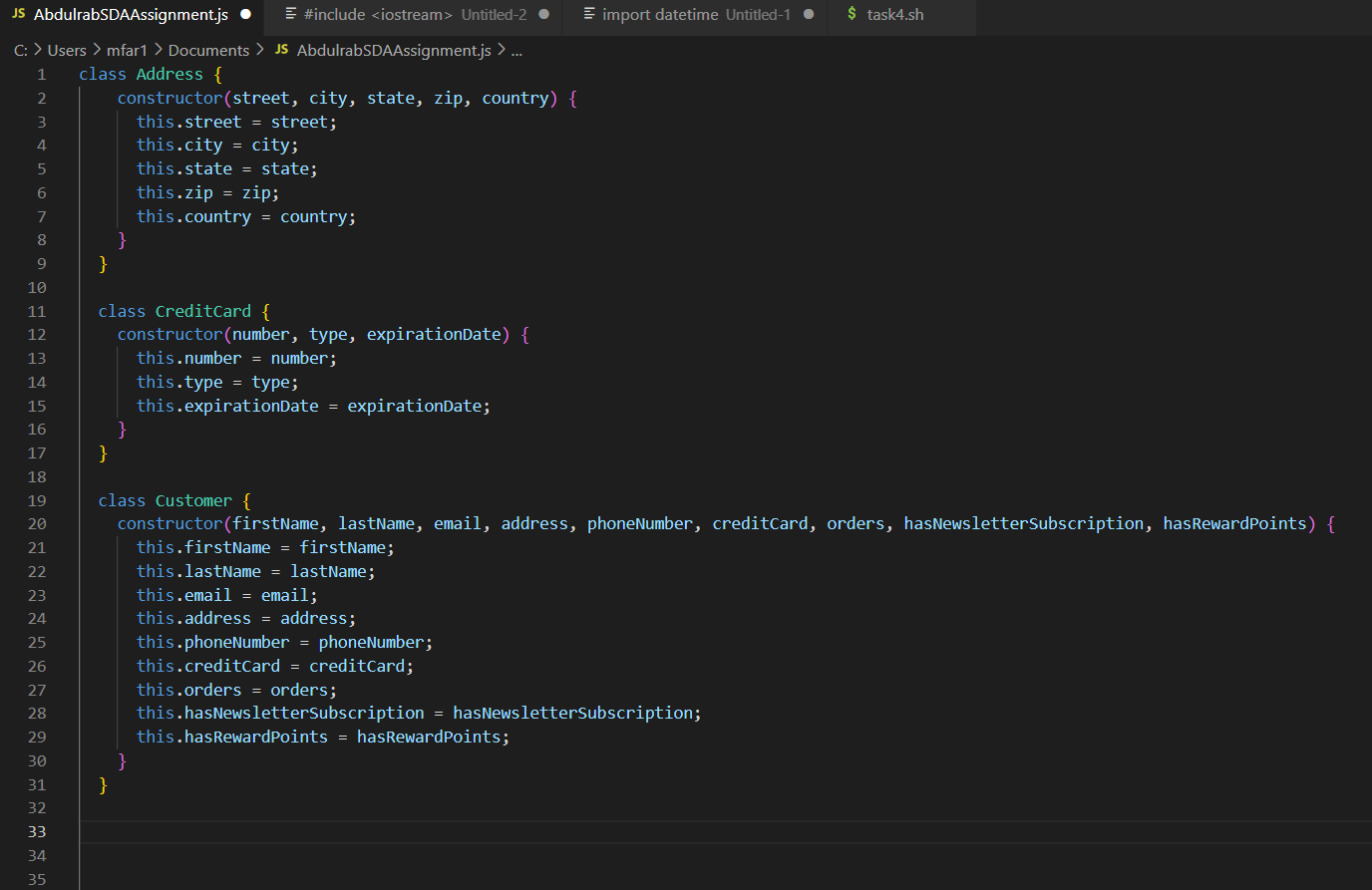
Description automatically generated



Text

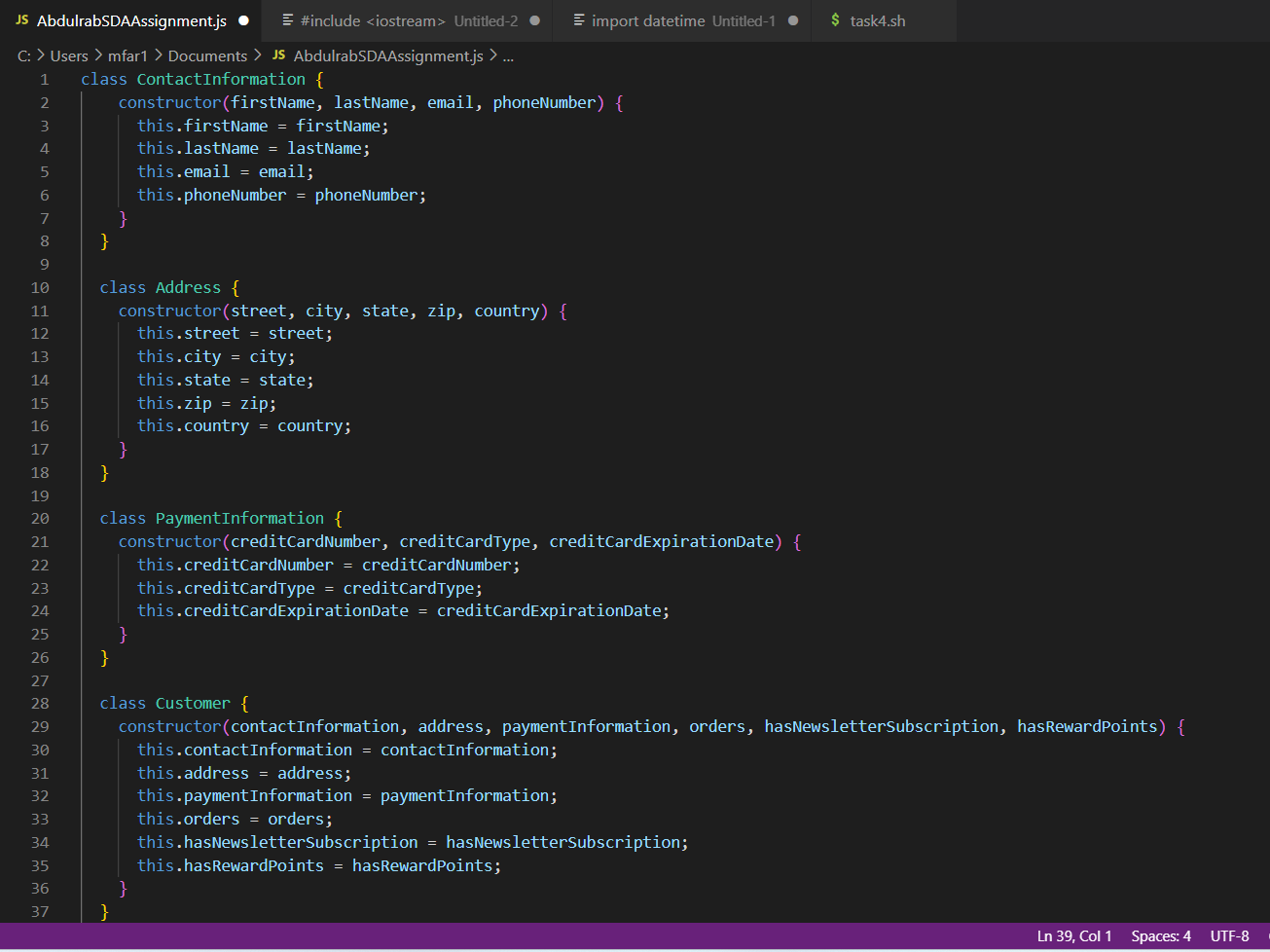
Description automatically generated

### Solution: Extract Class,



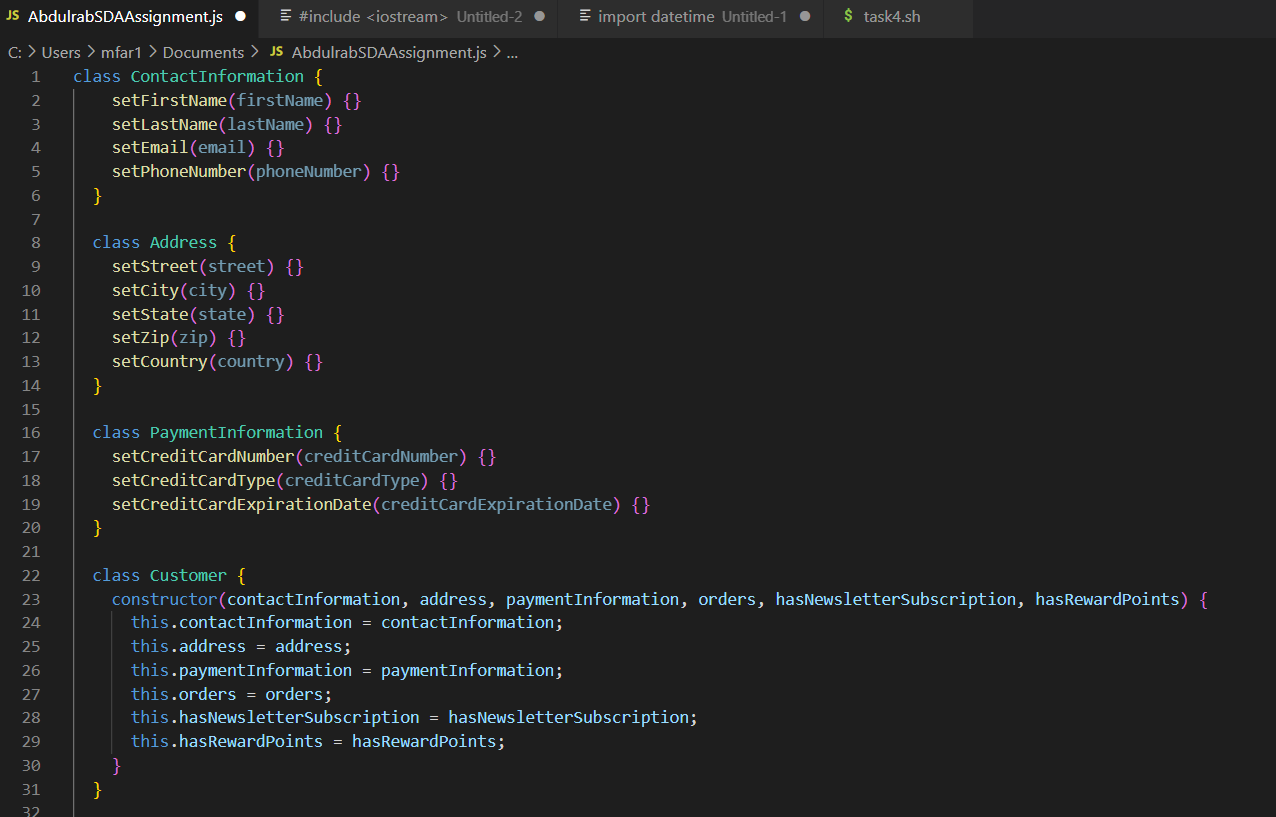
### Solution: Extract Subclass,

The Customer class now takes instances of the ContactInformation, Address, and PaymentInformation classes as arguments in its constructor, rather than individual properties for the customer's contact information, address, and payment information. This makes it easier to create new instances of the Customer class, and allows us to reuse the ContactInformation and PaymentInformation classes in other parts of our codebase.



### Solution: Extract Interface

In this refactored code, we extracted interfaces for the ContactInformation, Address, and PaymentInformation classes. Each interface defines a set of methods that the corresponding class must implement. The Customer class now takes instances of the interfaces as arguments in its constructor, rather than instances of the classes themselves. This allows us to decouple the Customer class from the implementation details of the ContactInformation, Address, and PaymentInformation classes. It also makes it easier to change the implementation of these classes in the future without affecting the Customer class. Note that in this refactored code, we didn't actually define the classes themselves, only the interfaces. This is because the implementation details of the classes are not important to the Customer class, as long as they implement the required methods.



### Solution: Duplicate Observed Data

Duplicating observed data is generally not considered good practice, as it can lead to inconsistencies and bugs in the code.