I look at building business as a creative process that I enjoy.

Chad Hurley

Group Member:	**Muhammad Amjad (181400181)**
	Shahana Parveen (181400122)
	Nor e Fatima (181400082)
Topic:	Refactoring
Course Title:	Software Reengineering
Course Code:	CS-368
Section:	**(B)**
Assignment #:	02
Assignment Date/Day:	April 9, 2022/Saturday
Submitted To:	**Sir Fakhar Lodhi**

Visual Studio Code Refactoring Support

Refactoring the source code will improve the quality and sustainability of your project by redesigning your code while not changing operating time behavior. Visual Studio Code supports the functionality of redesign as a Removal and Extract Variable strategy to enhance your code base within your editor.

For example, a common refactoring is used to avoid code duplication is an **Extract Method refactoring**, in which you select the source code you would like to use elsewhere and export it to its assigned path.

Refactorings are provided by the language service and the VS Code has built-in TypeScript and JavaScript refactoring support using the **TypeScript** language service. Rebuilding support for other programming languages is provided using VS **Code extensions** that provide language services. UI and redesign instructions are the same for all languages, and in this article we will show support for renewal with TypeScript language service.

Refactoring Apprauches in VS CODE

1. **Quick Fixes and refactorings**

In VS Code, Code Actions can provide both redesign and Quick Repair of detected problems (highlighted in green squiggles). Available Code Action is displayed by a lamp next to the source code when the cursor is in the moving or selected text area. By clicking on the Code Action lamp or using the Quick Fix command Ctrl +. will display Quick Repair and Reinstall.

If you would like to see a redesign without a quick fix, you can use the Refactor command (Ctrl + Shift + R).

2. **Refactoring actions**

2.1 Extract Methods

Select the source code you would like to extract and then press the lamp to rotate or press (Ctrl +.) To see available refactorings. Fragments of source code can be copied to a new path, or to a new task in a variety of different environments. During retrieval of the extract, you will be asked to provide an important name.

2.2 Extract Variable

TypeScript language service provides **Extract to const** refactoring to create a new local variable for the currently selected expression. When working with classes, you can also extract a value to a new property.

3. Rename Symbols

Renaming is a common function related to source code replication and VS Code has a separate Command Rename command (F2). Some languages support brand renaming in all files. Press F2 and type the new word you want and then press Enter. All use of the mark will be renamed, in all files

4. Keybindings for Actions

The editor.action.codeAction command lets you configure key binding for specific Code actions. This key binding, for example, activates the Copy Code Actions.

Code Action types are specified extensions using the advanced CodeActionProvided API. Genres are sequential, so "genre": "refactor" will show all code regenerative actions, and "genre": "refactor.extract.function" will only show re-releases.

By using the key binding above, if only one "refactor.extract.function" code is available, it will be used automatically. When Multiple Release Code

Actions are available, we bring up a context menu to select:

If the Code Action key obligation is set to "preferences": it is true, only Quick Instant and refactorings are shown. The Quick Fix option deals with the root error, while the preferred re-selection is the most frequently redesigned option. For example, although many refactor.extract.constant refactoring may be present, each extract in a different scope on file, the preferred refactor.extract.constant refactoring is the output of the local variable.

This key combination uses "preferences": it is true to create a re-enter that always tries to extract the selected source code into the local location.

5. Extensions with Refactoring

We can find extensions that support refactoring by looking in the VS Code Marketplace. Simply go to the Extensions view (Ctrl+Shift+X) and type 'refactor' in the search box. Also then sort by install count or ratings to see which extensions are popular.