**COMSATS UNIVERSITY OF ISLAMABAD,**

**ATTOCK CAMPUS**

**LAB TERMINAL**

**Submitted by: Ahmed Ali/Nadeem Mumtaz**

**Registration No: FA20-bcs-016/FA20-bcs-060**

**Class: BCS-7B**

**Submitted to: Sir Bilal Bukhari**

**Date: 27-12-2023**

**QUESTION NO 03:**

How functions work. Step by step

Form Initialization (`Form1\_Load`):

- Sets the initial size of the form. - Calls functions to populate lists for identifiers, symbols, and reversed words.

createIdentifiers():

- Populates the `iList` with different data types like "int", "float", "string", etc.

createSymbols():

- Creates `Symbols` objects and adds them to `sList`.

- Also populates `oList` with operator symbols.

createReversedWords():

- Populates `rList` with reversed words like "for", "while", etc.

createMemoryLabels():

- Dynamically creates labels to display memory and calculation data.

- Removes any previous labels and then adds the new ones.

printErrors():

- Displays errors in the code.

- The size of the form changes to accommodate the error display.

- Previous labels and memory displays are removed before displaying errors.

button1\_Click():

- Triggered when a button (possibly a compile or analyze button) is clicked.

- Splits the input code by spaces.

- Removes any previous labels and memory displays.

- Iterates over each tokenized component of the code and displays them with appropriate labels.

Initialization:

iList, sList, and rList are populated with predefined lists of identifiers, symbols, and reversed words respectively.

The createMemoryLabels function is designed to create labels based on memory and calculation lists, but its actual content is incomplete in the provided code.

Parsing Logic**:**When the button1\_Click event is triggered (presumably by pressing a button in the UI), the code inside this method is executed.

The entered code (textBox1.Text) is split into tokens using spaces.

For each token in the code:

It checks if the token is an identifier, symbol, reversed word, variable, number, pointer, or an error using various conditions and regular expressions.

Based on the type of token, a corresponding label is created and added to the flowLayoutPanel1.

Displaying Results**:**Once the tokens are processed, the labels are displayed in the UI, providing a visual representation of the parsed code.