COMSATS University Islamabad (CUI) Attock Campus

**C**

**Final Term**

**MUHAMMAD YAQOO CIIT/FA21-BCS-044/ATK**

**Submitted To: Dr. Bilal Haider  
subject: CC**

**Question1:**

Class Diagram:

+------------------+ +---------------------+

| Scanner | | Parser |

+------------------+ +---------------------+

| - components | | - syntaxTree |

| + Initialize() | | + Parse(tokens) |

| + Dispose() | | + GenerateTree() |

+------------------+ +---------------------+

| |

v v

+--------------------+ +------------------------+

| SemanticAnalyzer | | CodeGenerator |

+--------------------+ +------------------------+

| - symbolTable | | + GenerateCode() |

| + Analyze(tree) | | + OptimizeCode() |

+--------------------+ +------------------------+

|

v

+------------------+

| SymbolTable |

+------------------+

| - entries |

| + AddEntry() |

| + Lookup() |

+------------------+

+---------------------------+

| User Input |

+---------------------------+

|

v

+-----------------------------+

| Mini Compiler (Phases) |

| - Lexical Analysis |

| - Syntax Analysis |

| - Semantic Analysis |

| - Code Generation |

+-----------------------------+

|

v

+-------------------+

| Optimized Output |

**Question 2:**

**Explain two code functions of your mini compiler**

1. **Lexical Analyzer (Scan() Function)**:  
   This function reads the input source code, breaks it into tokens, and categorizes each token (keyword, identifier, operator, etc.).

csharp

Copy code

private void Scan()

{

string[] lines = textBox1.Text.Split('\n');

foreach (string line in lines)

{

// Tokenize the line

List<string> tokens = Tokenize(line);

// Display tokens in the output box

textBox2.AppendText(string.Join(" ", tokens) + "\n");

}

}

1. **Syntax Analyzer (Parse() Function)**:  
   This function checks whether the sequence of tokens forms a valid statement by applying grammar rules.

csharp

Copy code

private void Parse()

{

// Example: Basic grammar check for assignment statements

string input = textBox2.Text;

if (Regex.IsMatch(input, @"^[a-zA-Z\_]\w\*\s\*=\s\*\d+;$"))

{

MessageBox.Show("Syntax is correct");

}

else

{

MessageBox.Show("Syntax error");

}

}

**Question 3:**

**Examples of code optimizations (5)**

1. **Constant Folding**: Precomputing constant expressions during compilation.  
   Example: int a = 2 + 3; is optimized to int a = 5;.
2. **Dead Code Elimination**: Removing code that doesn’t affect the program's output.  
   Example: if (false) { ... } is entirely removed.
3. **Strength Reduction**: Replacing expensive operations with cheaper ones.  
   Example: x \* 2 becomes x + x.
4. **Inline Expansion**: Replacing function calls with the actual function code to reduce overhead.
5. **Loop Unrolling**: Expanding loops to reduce iteration overhead.