**Q4) Give input and output of the mini compiler.**

**Input:**

public class Program

{

public static void Main()

{

string code = @"

class Program

{

public static void Main()

{

// This is a comment

/\* This is a

multiline comment \*/

int x = 10;

double y = 5.5;

string message = ""Hello, World!"";

Console.WriteLine(message);

}

}

";

//string code = Console.ReadLine();

LexicalAnalyzer lexer = new LexicalAnalyzer(code);

List<Token> tokens = lexer.Analyze();

Console.WriteLine("--------------------Lexer Phase------------------");

foreach (var token in tokens)

{

Console.WriteLine(token);

}

// Step 2: Initialize the parser with the generated tokens.

Parser parser = new Parser(tokens);

// Step 3: Parse the tokens to generate the Abstract Syntax Tree (AST).

AST ast = parser.Parse();

// Step 4: Display the AST or handle parsing errors.

if (ast != null)

{

Console.WriteLine();

Console.WriteLine("--------------------Parser Phase------------------");

Console.WriteLine("AST:");

ast.Print();

SemanticAnalyzer semantic = new SemanticAnalyzer();

semantic.Analyzer(ast);

semantic.printSymbolTable(ast);

ThreeAddressCode three = new ThreeAddressCode();

var threeAdressCode = three.generateThreeAddressCode(ast);

CodeOptimization optimizer = new ThreeAddressCode();

var optimized = optimizer.optimize(threeAddressCode);

}

else

{

Console.WriteLine("Parsing failed.");

}

Console.ReadKey();

}

}

**Output:**



