CO523 – Programming Languages

Assignment 01: Execution Models & Language Processing

Total Marks: 100
Instructions:

- Answer all questions.
- Use clear explanations and diagrams where appropriate.
- Programming examples may be written in C, Java, or Python unless stated otherwise.

Section A – Short Answer Questions (10 × 4 = 40 marks)

Answer **all** questions briefly.

1. Define an **execution model** in the context of programming languages.
2. Differentiate between **compiled**, **interpreted**, and **hybrid** execution models with one example each.
3. What is a **Virtual Machine (VM)**? State two advantages of using a VM.
4. List and briefly explain the components of a **runtime environment** of a program.
5. Define **lexical analysis**. What are the **input** and **output** of a lexical analyzer?
6. What is a **token**? List any four types of tokens with examples.
7. Explain the difference between a **lexeme** and a **token**.
8. What is **syntax analysis**? What kind of errors are detected during this phase?
9. Define a **symbol table** and mention two types of information stored in it.
10. Explain the analogy between **lexical analysis & syntax analysis** and natural language processing.

Section B – Descriptive Questions (4 × 10 = 40 marks)

Answer **any four** questions.

11. Explain the **program execution process** from source code to execution, clearly describing the role of:

- Compiler / Interpreter
- Virtual Machine
- Runtime Environment

12. Describe the **lexical analysis phase** in detail. Explain its responsibilities and illustrate the process using the following statement:

```
float average = total / count;
```

13. Explain **syntax analysis** with the help of a suitable example. Describe how a **parse tree** is generated and how syntax errors are detected.
14. Compare **static (lexical) scoping** and **dynamic scoping**. Use code examples to justify your answer.
15. Explain different **variable lifetimes** (static, stack-dynamic, heap-dynamic) and relate them to memory regions (stack, heap, static memory).
16. Discuss **memory management techniques** in programming languages.
    Compare **manual memory management** and **automatic garbage collection** with examples.

Section C – Application / Analytical Questions (2 × 10 = 20 marks)

Answer **any two** questions.

17. Given the following code snippet:

```
int x = 5;
x = x + 1;
```

a) List all **tokens** generated by the lexical analyzer.
b) Identify the **token type** for each token.
c) State whether any lexical error exists. Justify your answer.

18. Consider the following incorrect statement:

```
int = value 20;
```

a) Will the lexical analyzer detect an error? Why?
b) Will the syntax analyzer detect an error? Explain clearly.
c) Identify the phase responsible for reporting this error.

19. Explain how a **symbol table** is used during program execution. Illustrate your answer by showing sample symbol table entries for a simple function.
20. Analyze how **modern programming languages** (e.g., Python, Rust, Go) improve safety and reliability using concepts such as:

- Memory safety
- Type safety
- Concurrency safety