

CO523 – Programming Languages

Assignment 02: Type System Study

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.
-

A **type system** defines how a programming language classifies values and expressions into types and enforces constraints on their usage. Type systems help detect errors, improve readability, and enhance software reliability.

Programming languages differ significantly in how strictly they enforce type rules. Understanding **strong vs weak typing** and **static vs dynamic type checking** is essential for designing safe and efficient software systems.

This assignment has four Components

Task 1 – Data Types & Type Systems

Task 2 – Type Checking

Task 3 – Strong vs Weak Typing

Task 4 – Case Study

Task 1: Data Types and Type Systems (60 minutes)

(a) Data Types

Explain the following types with examples from **at least two programming languages** (e.g., C, Java, Python):

- Primitive data types
- Composite data types
- User-defined data types

(b) Type Systems

Define a **type system** and explain its purposes, including:

- Error detection
- Program correctness
- Compiler optimizations

with code examples.

Task 2: Type Checking Mechanisms (60 minutes)

(a) Static Type Checking Write Concept explanation + code examples.

- Definition
- Advantages and disadvantages
- Languages that use it

(b) Dynamic Type Checking Write Concept explanation + code examples.

- Definition
- Advantages and disadvantages
- Languages that use it

(c) Code Illustration

Provide **one static** and **one dynamic typing** example.

Write Concept explanation + code examples.

Task 3: Strong vs Weak Typing (90 minutes)

(a) Definitions

Explain:

- Strong typing
- Weak typing

(b) Comparative Analysis

Compare **strong vs weak typing** using the following languages:

Feature	C	Java	Python	JavaScript
Typing Strength				
Type Conversion				
Implicit Coercion				
Safety Level				

(c) Code Examples

Show **implicit type coercion** in weakly typed languages and **type safety** in strongly typed languages. Write Table + explanation + annotated code.

Task 4: Case Study and Discussion (450 minutes)

Scenario:

You are designing:

- A **banking system**
- A **scientific simulation**
- A **web application**

Questions:

1. Which type system would you prefer (strong/weak, static/dynamic)?
2. Justify your choice based on:
 - Safety
 - Performance
 - Maintainability

Write Well-structured discussion (3000–4000words).

4. Submission Guidelines

- Format: **PDF / DOCX**
- Code language: C / Java / Python / JavaScript
- Diagrams encouraged (optional)
- Plagiarism policy applies