University of Jijel **Master AI**

Faculty of Exact Sciences and Computer Sci. **Programming Language Semantics**

Dept. of Computer Science **Project 02**

Reasoning About Programs with Assertions

In this homework, you will explore ideas from axiomatic semantics and Hoare logic through concrete programming experiments. Instead of working with formal proofs, you will reason about program correctness by annotating JavaScript programs with assertions that describe what should be true before, during, and after execution.

By writing and testing assertions, you will experience the role of preconditions, postconditions, and loop invariants, and understand both their power and their limitations. This assignment focuses on using code to support reasoning about programs, and on discovering why proving correctness is harder than testing.

Work to Do:

1. A Small Imperative Language (in JavaScript)

Write several small JavaScript programs using:

- Variable assignments
- Conditionals (if)
- Loops (while)

Choose programs simple enough to understand fully, but complex enough to require careful reasoning.

1. Program Annotations

For each program, write annotations in comments:

```
// Precondition:
// Loop invariant:
// Postcondition:
```

- Preconditions describe assumptions about inputs.
- Invariants describe what must remain true at each loop iteration.
- Postconditions describe the expected result.
    1. Runtime Assertions

Translate your annotations into runtime checks using assertions:

```
console.assert(condition, "error message");
```

Insert assertions:

- Before the program starts
- At the beginning (or end) of each loop iteration
- After the program finishes