LAB EXERCISE 5

PART 1: DENOTATIONAL SEMANTICS

Problem 1: Define denotational semantics for a simple programming language with arithmetic expressions, Boolean expressions, and conditional statements (if-else).

Problem 2: Consider a language with arithmetic expressions involving addition, subtraction, multiplication, and division. Define denotational semantics for this language.

Problem 3: Define denotational semantics for a programming language that includes Boolean expressions with AND, OR, and NOT operators, along with conditional statements.

Problem 4: Define denotational semantics for a language that includes arithmetic expressions, Boolean expressions, conditional statements, and loops (while loops).

PART 2: OPERATIONAL SEMANTICS

Problem 5: Define small-step operational semantics for a simple programming language with arithmetic expressions and assignment statements.

Problem 6: Consider a language with Boolean expressions involving AND, OR, and NOT operators, along with conditional statements. Define small-step operational semantics for this language.

Problem 7: Define small-step operational semantics for a language that includes arithmetic expressions, Boolean expressions, conditional statements, and loops (while loops).

Problem 8: Define big-step operational semantics for a programming language with arithmetic expressions, Boolean expressions, and conditional statements.

Problem 9: Define big-step operational semantics for a language that includes arithmetic expressions, Boolean expressions, conditional statements, loops (while loops), and function definitions.

Problem 10: Consider a language with recursive function definitions. Define big-step operational semantics for this language.

Problem 11: Define operational semantics for a language that includes features like exception handling (try-catch blocks).

Problem 12: Define operational semantics for a language that includes features like object-oriented programming constructs (classes, objects, inheritance).