Name: _____ **CUNY Id:** _____

Queens College of CUNY
Department of Computer Science
Principles of Programming Languages
(CSCI 316)
Winter 2026
Midterm Exam - Part A

Instructions:

- *This is a closed-book exam. No printed resources (notes, slides, books, etc.) nor any electronic devices (smart watches, phones, calculators, tablets, laptops, desktops, etc.) are permitted. Use of forbidden resources or communication with others (beside the instructor/proctor) during the exam will result in confiscation of the exam and referral of the case to the appropriate department/college offices for further handling.*
- *You must do all eight (8) questions, worth thirteen (13) points each. Part A, contains Problems 1-4. After you submit Part A, you can take a brief break not to exceed five minutes; upon your return, you will be given Part B which contains Problems 5-8. Part B must be submitted by 3:30 p.m. It is recommended that Part A be submitted no later than 2:15 p.m. Bathroom and other breaks can be taken only* between *Parts A and B, not during either part.*
- *Partial credit is awarded, so please attempt each question even if you cannot complete it in its entirety.*

Problem:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**TOTAL (Exam)**	**TOTAL (Quizzes 0-7)**	**TOTAL (Assns 1-7)**
Score:											

Problem 1: Chomsky Hierrachy

[a] Complete this chart about the Chomsky Hierarchy, a classification of formal languages and a foundation of the Theory of Computation course (CSCI 320) which is a prerequisite of our course Principles of Programming Languages (CSCI 316).

Type	Grammar	Language	Automation	Production Rules
Type-0				
Type-1				
Type-2				
Type-3				

[b] Describe which of these Grammars / Languages / Automata is/are most relevant to our Programming Languages course, and in what way?

Problem 2: Language Features (13 x 1 point = 16 points)

Identify the language of each code snippet and explain what feature(s) led you to that identification.

(Choices are: Algol, APL, C, C++, Cobol, Fortran, Java, Javascript Lisp, Pascal, PHP, PL/I, Plankalkül, Prolog, Python, Ruby, Short Code, SNOBOL)

```
    |  A + 1 => A
V   |  4        5
 S  |  1.n      1.n
```

```
declare i fixed binary(31);
declare total float;
do i = 1 to 10;
   total = total + i;
end;
```

```
01 -    06 abs value
02 )    07 +
03 =    08 pause
04 /    09 (
```

```
line = INPUT
pat  = SPAN('0123456789')
line pat = number :S(FOUND)
OUTPUT = 'No number found'
FOUND OUTPUT = number
```

```
real :: a(5), b(5)
a = (/ 1, 2, 3, 4, 5 /)
b = a * a
print *, b
end
```

```
A ← 1 2 3 4 5
B ← A × A
□ ← B
```

```
(defun sum-list (lst)
  (if (null lst)
      0
      (+ (car lst) (sum-list (cdr lst)))))
```

```
01  CUSTOMER-RECORD.
    05 CUSTOMER-ID      PIC 9(5).
    05 CUSTOMER-NAME    PIC X(20).
    05 ACCOUNT-BALANCE  PIC 9(7)V99.
```

```
procedure fact(n);
integer n;
if n = 0 then 1
else n * fact(n - 1);
```

```
var
  i, sum: integer;
begin
  for i := 1 to 10 do
    sum := sum + i;
end.
```

```
struct Point { int x, y; };
struct Point p = {3, 4};
struct Point *q = &p;
q->x += 1;
printf("%d\n", p.x);
```

```
$total = 0;
foreach ($_GET as $key => $value)
    $total += (int)$value;
echo $total;
```

```
public class Main {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 1; i <= 10; i++) sum += i;
        System.out.println(sum);
    }
}
```

```
#include <iostream>
struct Counter {
    int x; Counter(int v) : x(v) {}
    ~Counter() { std::cout << x << std::endl; }
};
```

```
const user = { name: "Alice" };
user.age = 30;
user.greet = function () {
  console.log("Hi " + this.name);
};
```

```
def main():
    total = sum(range(1, 11))
    print(total)

if __name__ == "__main__":
    main()
```

```
edge(a, b).
edge(b, c).
path(X, Y) :- edge(X, Y).
path(X, Y) :- edge(X, Z), path(Z, Y).
```

```
sum = 0
(1..10).each do |i|
  sum += i
end
puts sum
```

Problem 3: Preliminary Principles

[a] Give three reasons for studying the principles of programming languages

[b] State and describe the four core evaluation criteria for programming languages.

[c] Define these terms, related to the evaluation criteria mentioned above:

Orthogonality -

Aliasing -

Portability -

Generality -

[d] What is the "Von Neumann" computer architecture model and how is it related to study of programming languages.

Problem 4: Variables

[a] Name and explain the six major attributes of variables.

Attribute	Explanation

[b] Name and explain at least three different kinds of "scope":

Type	Explanation

Name: _____ **CUNY Id:** _____

Principles of Programming Languages
(CSCI 316)
Winter 2026
Midterm Examination - Part B

Problem 5: Lexical and Syntax Analysis

[a] Draw a finite-state machine to recognize an acceptable variable name in PHP, subject to the following rules:

- It must begin with a dollar sign ($)
- The dollar sign ($) must be followed by a letter (A-z) or an underscore (_).
- The subsequent characters must be alphanumeric (A-z, 0-9) or underscores (_)
- There are other rules, but they have been omitted to simplify the problem.

[b] Consider the following production rules of a context-free grammar.

<program> → <statement>
<statement> → <assignment>
<assignment> → <identifier> = <expression>
<expression> → <identifier> + <number> | <identifier> + <identifier>
<identifier> → x | y
<number> → <digit> | <digit> <number>
<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Provide a "derivation" of the statement "y = x + 26".

Problem 6: Expressions and Assignments

Explain each of these concepts relating to expressions and assignments and give a specific issue which must be addressed by the designers and implementers of a programming language and the different ways that the issue may be handled.

Concept	Explanation	Issues and Approaches
Arithmetic Expression		
Overloaded Operator		
Type Conversion		
Relational and Boolean Expressions		
Short-Circuit Evaluation		
Assignment Statements		
Mixed-Mode Assignments		

Problem 7: Data Types (2 points each, max 13 points)

[a] Name and explain two ways in which a negative signed integer may be stored.

[b] Name and explain how a floating-point number is stored (including how it differs from how an integer is stored)

[c] Explain the terms "precision" and "range"

[d] What major issue is there with floating points and what is an alternative that might be used in a business application

[e] What is string slicing and what are some of the issues an options.

[f] What concern might there be with code like strcpy(dest, src)?

[g] For two-dimensional arrays, what do the terms "jagged array" and "column-major order" respectively mean?
.

[h] What is an "associative array" and how does it differ from a conventional one?

[i] What are two approaches to supporting additional data types that might be needed in a programming language?

Problem 8: Syntax and Semantics

Each box below is worth one point. Fill 13 points worth.

[a] Explain the following terms pertaining to "Syntax"

Type	Explanation / Illustration
Lexeme	
Token	
Sentimental Form	
Sentence	
Ambiguous	
BNF	
LL Parser	
LR Parser	
Terminal	

[b] Name, explain, and illustrate the three major types of "Semantics" that we studied:

Type	Explanation	Illustrative Example