Queens College of CUNY
Department of Computer Science
Programming Languages
(CSCI 316)
Winter 2026

Midterm Info

The Midterm Exam will take place on Tuesday, January 13 in our classroom SB B145.

The material for the exam is Chapters 1-7 (even if we likely do Chapter 8 before the Midterm). You are also responsible for takeaways from our assignments. (While you are not expected to write some of the complex programs used in class, you should understand the outputs that we discussed and analyzed.)

There will be eight (mult-part) questions, worth 13 points each, with approximately one associated with each chapter.

The question on Chapter 2 - "Evolution of Major Programming Languages", but also related to the course as a whole, will give you representative code snippets in various languages and ask you to identify the language and what feature(s) you observed that led to that identification.

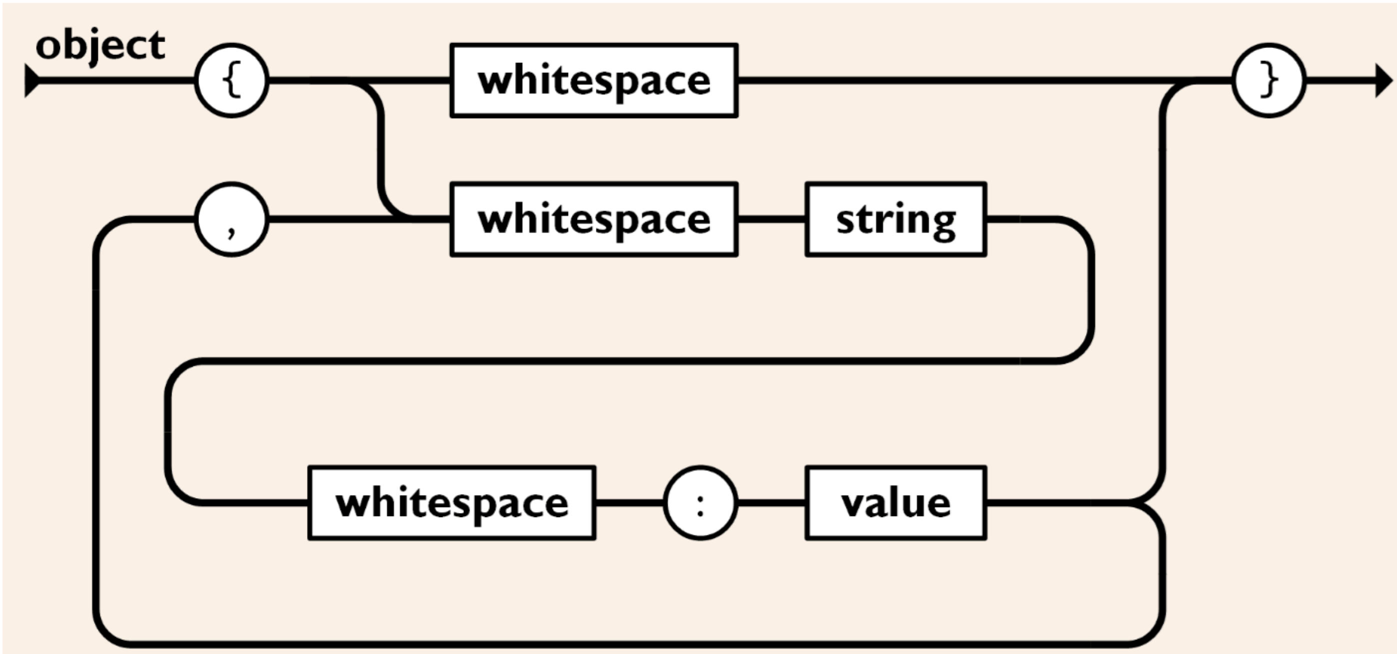
There will also be a question on the Chomsky hierarchy of "formal languages" and knowing which types are most applicable to our study of "programming languages" and why.

Grammar	Languages	Recognizing automaton	Production rules (constraints)[a]	Examples[5][6]
Type-0	Recursively enumerable	Turing machine	$\gamma \rightarrow \alpha$ (γ non-empty)	$L = \{w \mid w$ describes a terminating Turing machine $\}$
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$	$L = \{a^n b^n c^n \mid n > 0\}$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \rightarrow \alpha$	$L = \{a^n b^n \mid n > 0\}$
Type-3	Regular	Finite-state automaton	$A \rightarrow \mathrm{a}$ $A \rightarrow \mathrm{a}B$ (right regular) or $A \rightarrow \mathrm{a}$ $A \rightarrow B\mathrm{a}$ (left regular)	$L = \{a^n \mid n > 0\}$

- Type-0 grammars, known as "Unrestricted Grammars": $(V \cup \Sigma)^+ \rightarrow (V \cup \Sigma)^*$
- Type-1 grammars known as "Context-Sensitive Grammars" : $V \cup \Sigma)^+ \rightarrow (V \cup \Sigma)^+$ - always increasing in size
- Type-2 grammars, known as "Context-Free Grammars" (CFG): $V \rightarrow (V \cup \Sigma)^*$
- Type-3 grammars, known as "Regular Grammars": $V \rightarrow \Sigma \mid \Sigma V$

* i"Kleene closure" means 0 or more copies
+ "positive closure" means 1 or more copies

You should also be familiar with the state-diagrams used to describe language-constructs (e.g. JSON) and how they relate to the production rules of grammars.



<object> → { <whitespace> } |
{ <whitespace> <key-value-pair> } |
{ <whitespace> <key-value-pair> <kv-pair-var>}
<kv-pair-var>→ , <whitespace> <key-value-pair> <kv-pair-var> | λ
<key-value-pair> → <string> <whitespace> : <value>
<whitespace> →