Try the following exercises. Compute the First and Follow sets as well as construct the parsing table for the following LL(1) grammars. Solutions are on the next page.

Exercise 1

S $\rightarrow$ A a
A $\rightarrow$ B D
B $\rightarrow$ b | ϵ
D $\rightarrow$ d | ϵ

Exercise 2

C $\rightarrow$ P F class id X Y
P $\rightarrow$ public | ϵ
F $\rightarrow$ final | ϵ
X $\rightarrow$ extends id | ϵ
Y $\rightarrow$ implements I | ϵ
I $\rightarrow$ id J
J $\rightarrow$, I | ϵ

Exercise 3

prog $\rightarrow$ stmt
stmt $\rightarrow$ if expr then block | while expr do block | expr ;
expr $\rightarrow$ term $=>$ id | isZero? term | not expr | $++$ id | $--$ id
term $\rightarrow$ id | const
block $\rightarrow$ stmt | { stmts }
stmts $\rightarrow$ stmt stmts | ϵ

Exercise 1

1. S $\rightarrow$ A a
2. A $\rightarrow$ B D
3. B $\rightarrow$ b
4. B $\rightarrow \epsilon$
5. D $\rightarrow$ d
6. D $\rightarrow \epsilon$

First(S) = {b, d, a}
First(A) = {b, d, ϵ}
First(B) = {b, ϵ}
First(D) = {d, ϵ}

Follow(S) = {$}
Follow(A) = {a}
Follow(B) = {d, a}
Follow(D) = {a}

	a	b	d	$
S	1	1	1	
A	2	2	2	
B	4	3	4	
D	6		5	

Exercise 2

1. C $\rightarrow$ P F class id X Y
2. P $\rightarrow$ public
3. P $\rightarrow \epsilon$
4. F $\rightarrow$ final
5. F $\rightarrow \epsilon$
6. X $\rightarrow$ extends id
7. X $\rightarrow \epsilon$
8. Y $\rightarrow$ implements I
9. Y $\rightarrow \epsilon$
10. I $\rightarrow$ id J
11. J $\rightarrow$, I
12. J $\rightarrow \epsilon$

First(C) = {public, final, class}
First(P) = {public, ϵ}
First(F) = {final, ϵ}

First(X) = {extends, ϵ}
First(Y) = {implements, ϵ}
First(I) = {id}
First(J) = {',', ϵ}

Follow(C) = {$}
Follow(P) = {final, class}
Follow(F) = {class}
Follow(X) = {implements, $}
Follow(Y) = {$}
Follow(I) = {$}
Follow(J) = {$}

	public	final	class	id	extends	implements	,	$
C	1	1	1					
P	2	3	3					
F		4	5					
X					6	7		7
Y						8		9
I				10				
J							11	12

Exercise 3

1. prog $\rightarrow$ stmt
2. stmt $\rightarrow$ if expr then block
3. stmt $\rightarrow$ while expr do block
4. stmt $\rightarrow$ expr ;
5. expr $\rightarrow$ term => id
6. expr $\rightarrow$ isZero? term
7. expr $\rightarrow$ not expr
8. expr $\rightarrow$ ++ id
9. expr $\rightarrow$ -- id
10. term $\rightarrow$ id
11. term $\rightarrow$ const
12. block $\rightarrow$ stmt
13. block $\rightarrow$ { stmts }
14. stmts $\rightarrow$ stmt stmts
15. stmts $\rightarrow$ ϵ

First(prog) = {if, while, id, const, isZero?, not, ++, --}
First(stmt) = {if, while, id, const, isZero?, not, ++, --}
First(expr) = {id, const, isZero?, not, ++, --}
First(term) = {id, const}

First(block) = {'{', if, while, id, const, isZero?, not, ++, −−}
First(stmts) = {ϵ, if, while, id, const, isZero?, not, ++, −−}

Follow(prog) = {$}
Follow(stmt) = {$, if, while, id, const, isZero?, not, ++, −−}
Follow(expr) = {then, do, ;}
Follow(term) = {=>, then, do, ;}
Follow(block) = {$, if, while, id, const, isZero?, not, ++, −−}
Follow(stmts) = {'}'}

	if	while	id	const	isZero?	not	++	−−	{	then	do	;	=>	}	$
prog	1	1	1	1	1	1	1	1							
stmt	2	3	4	4	4	4	4	4							
expr			5	5	6	7	8	9							
term			10	11											
block	12	12	12	12	12	12	12	12	13						
stmts	14	14	14	14	14	14	14	14						15	