CS 2210 – Homework 2 Solutions

1.)

S → aB

B → bB | a

2.)

Without precedence:

E → E + E | E - E | E *E | E / E | int

We want + and - to be higher in the parse tree, so we add a new nonterminal for E to go to:

E → E + T

E → E - T

E → T

T → T * F

T → T / F

T → F

F → int

We can check that we can derive the string to parse in 4) from this grammar:

E ⇒ **E** - T ⇒ **E** + T - T ⇒ **T** + T - T ⇒ **T** * F + T - T ⇒ **F** * F + T - T

⇒ int * **F** + T - T ⇒ int * int + **T** - T ⇒ int * int + **T** * F - T

⇒ int * int + **F** * F - T ⇒ int * int + int * **F** - T

⇒ int * int + int * int - **T** ⇒ int * int + int * int - **F**

⇒ int * int + int * int - int

This is a leftmost derivation, but we need still need to change the grammar since there is left recursion. We replace it with right recursion.

E → T E′

E′ → + T E′

E′ → - T E′

E′ → ε

T → F T′

T′ → * F T′

T′ → / F T′

T′ → ε

F → int

3.)

	First	Follow
-	-	N/A
+	+	
*****	*	
/	/	
int	int	
E	int	$
E'	+, -, ε	$
T	int	+, -, $
T'	*, /, ε	+, -, $
F	int	*, /, +, -, $

LL(1) Parse Table

	int	+	-	*	/	$
E	E → T E'					
E'		E' → + T E'	E' → - T E'			E' → ε
T	T → F T'					
T'		T' → ε	T' → ε	T' → * F T'	T' → / F T'	T' → ε
F	F → int					

4.)

Stack	Input	Action
E $	int * int + int * int - int $	E → T E’
T E’ $	int * int + int * int - int $	T → F T’
F T’ E’ $	int * int + int * int - int $	F → int
int T’ E’ $	int * int + int * int - int $	**terminal**
T’ E’ $	* int + int * int - int $	T’ → * F T’
*** F T’ E’ $**	* int + int * int - int $	**terminal**
F T’ E’ $	int + int * int - int $	F → int
int T’ E’ $	int + int * int - int $	**terminal**
T’ E’ $	+ int * int - int $	T’ → ε
E’ $	+ int * int - int $	E’ → + T E’
+ T E’ $	+ int * int - int $	**terminal**
T E’ $	int * int - int $	T → F T’
F T’ E’ $	int * int - int $	F → int
int T’ E’ $	int * int - int $	**terminal**
T’ E’ $	* int - int $	T’ → * F T’
*** F T’ E’ $**	* int - int $	**terminal**
F T’ E’ $	int - int $	F → int
int T’ E’ $	int - int $	**terminal**
T’ E’ $	- int $	T’ → ε
E’ $	- int $	E’ → - T E’
- T E’ $	- int $	**terminal**
T E’ $	int $	T → F T’
F T’ E’ $	int $	F → int
int T’ E’ $	int $	**terminal**
T’ E’ $	$	T’ → ε
E’ $	$	E’ → ε
$	$	**accept**