

Name: _____

Student# _____

2) Define the following terms:

- a) Concrete Syntax Tree
- b) Abstract Syntax Tree
- c) Semantic Error

Signature: _____

Name: _____

Student# _____

3) Design abstract syntax tree nodes for the following grammar

program	→	vardec* statement*
vardec	→	type ID ;
type	→	INT
	→	REAL
statement	→	print (exp) ;
	→	ID = exp ;
exp	→	exp '+' exp
	→	exp '-' exp
	→	INTEGER_LITERAL
	→	REAL_LITERAL
	→	ID
	→	(exp)

Signature: _____

Name: _____

Student# _____

4) Consider the following MiniJava code:

```
class Fibonacci{
    public static void main(String[] a){
        System.out.println(new Fib().ComputeFib(10));
    }
}

class Fib {

    public int ComputeFib(int num){
        int num_aux;

        if (num < 2)
            num_aux = num;
        else num_aux = this.ComputeFib(num-1) +
                       this.ComputeFib(num-2) ;
        return num_aux;
    }
}
```

Sketch the symbol table for the above code.

Signature: _____

Name: _____

Student# _____

5) Draw in detail how the program stack as well as the stack frames of the last calling and the called functions look like once the bold formatted call to f is executed. Show also where the frame and stack pointers point to in the stack frame.

```
int f(int a, int b, int c){
    int d, e;

    ...
    f(c+1, d+2, e);
    ...
}
```

Signature: _____

Name: _____

Student# _____

6) Consider the following IR syntax:

```
CONST(exp)
BINOP(op, exp1, exp2)
MEM(exp)
CJUMP(exp, true_lbl, false_lbl)
JUMP(lbl)
MOVE(mem_addr_exp, exp)
MOVE(temp_reg, exp)
LABEL(lbl_name)
```

Lower the following assign statement to the above provided IR syntax. Assume variables a and b are declared locally inside the enclosing method and c is a class variable.

```
int a, b;
...
a = b + this.c;
```

Signature: _____