Programming Languages

Final Exam

Name:____________

Answer all *seven* questions.

Question 1

Give a regular expression for each of the following languages over $\Sigma = \{0, 1, \ldots, 9\}$.

1. All 5-digit positive integers.
2. All positive integers that begin with 9 and are multiples of 5.
3. All strings that begin with 9 and contain three consecutive 1's.

Question 2

Prove that the following grammar is ambiguous.

```
E  -> E or E | E and E | not E | ( E ) | x
```

Question 3

What is dynamic binding? Give a example in Java or any other OOP language to illustrate it.

Question 4

Implement a class in C++ or Java named `MyCollection` that extends a collection class such as the vector or the linked list class. This new class overrides the method `add` in the following way: it does nothing if the object to be added already exists in the collection; otherwise it calls the `add` method in the super class to add the object into the collection. Use `o1.equals(o2)` to test if two objects `o1` and `o2` are equal.

Question 5

Write a function in Python, Haskall, or Picat to remove duplicates from a given list. For example, if the list contains 1, 1, 2, 3 and 2, then the resulting list should contain 1, 2, and 3. The order must be preserved. If possible, implement a $O(n)$ or $O(n \times log_2(n))$ algorithm, where n is the size of the given list.

Question 6

Assume an integer set is represented as an ordered list of integers without duplicates. Define the following functions on integer sets in Python, Haskall, or Picat.

1. `union(S1,S2)`: the union of `S1` and `S2`.
2. `exclusive_or(S1,S2)`: the "exclusive or" of `S1` and `S2`, i.e., the elements in `S1` or `S2` but not in both.
3. `power(S)`: The power set of `S`.

Question 7

In Picat, a binary tree can be represented as a structure in the form `t(Value,Left,Right)`, where `Left` is the left subtree and `Right` is the right subtree. An empty tree is represented as the atom `void`. Consider the following functions:

```
f1(void) = 0.
f1(t(_,Left,Right)) = N =>
    N = f1(Left) + f1(Right) + 1.

f2(void) = [].
f2(t(Value,void,void)) = [Value].
f2(t(_,Left,Right)) = L =>
    L = f2(Left) ++ f2(Right).
```

1. What is the result of each of the following function calls?
   (a) `f1($t(1,void,void))`
   (b) `f1($t(1,t(2,void,void),t(3,void,void)))`
   (c) `f2($t(1,void,void))`
   (d) `f2($t(1,t(2,void,void),t(3,void,void)))`
2. Rewrite `f1` and `f2` to make them tail-recursive.