Answer any 5 of the following 6 problems. You may attempt all 6 for extra credit. (All the questions are equally weighted.)

1. Compilation
   (a) Explain the difference between a compiler and an interpreter.
   (b) Java has the properties of both a compiler and an interpreter. Briefly explain how this is so.
   (c) What is the purpose of a frame pointer? Describe how it works, and when it is set over the course of a function call and return.

2. Datatypes
   (a) What is an abstract data type? Why are they important?
   (b) Identify the 3 parts that make up the specification of a data type and give a short explanation for each.

3. Program Verification
   (a) Given \( \{P\} S \{Q\} \) for a pre-condition \( P \), post-condition \( Q \) and program \( S \), define the program’s partial correctness and total correctness.
   (b) Consider the following figure where \( S \) is a non-deterministic program and \( P_1, P_2 \) and \( Q \) are predicates.

\[ WP(S, Q) \]

\[ P_1 \]

\[ P_2 \]

i. For a state \( S_1 \in P_1 \), is it possible that execution of program \( S \) starting from \( S_1 \) will terminate outside \( Q \)? Explain.

ii. For a state \( S_2 \in P_2 \), is it possible that execution of program \( S \) starting from \( S_2 \) will terminate in \( Q \)? Explain.

iii. For a state \( S_1 \in P_1 \), is it possible that execution of program \( S \) starting from \( S_1 \) will not terminate? Explain.

iv. For a state \( S_2 \in P_2 \), is it possible that execution of program \( S \) starting from \( S_2 \) will not terminate?

(c) Write a program that, given a fixed integer \( n > 0 \), sets variable \( i \) to the highest power of 2 that is at most \( n \). The precondition \( Q \), postcondition \( R \), loop invariant \( P \), and bound function \( t \) are

\[ Q : \quad 0 < n \]

\[ R : \quad 0 < i \leq n < 2 \cdot i \land (\exists p : i = 2^p) \]

\[ P : \quad 0 < i \leq n \land (\exists p : i = 2^p) \]

\[ t : \quad n - i \]
(d) Construct a program to compute any possible index, \( \text{maxIndex} \), of an array \( A[0..N] \) such that \( A[\text{maxIndex}] \) is a maximum value. Precondition is: \( N \geq 0 \) Postcondition is:
\[
\{0 \leq \text{maxIndex} \leq N\} \text{ and } \{\forall \text{index} : 0 \leq \text{index} \leq N : A[\text{maxIndex}] \geq A[\text{index}]\}
\]
Formally prove that your program is totally correct (partial correctness and termination).

4. Objects

Given the program:

```java
public class A extends B {
    public int a;
    public double b;
    public char c;

    public A() {
    }

    public A(int a, double b, char c) {
    }

    public B(A a) {
    }
}
```

Note that there is single inheritance only!

(a) Diagram the structure of the object as it will be configured in memory.
```
A a = new A();
```

(b) In what part of memory would the object be allocated?

5. Parameter Passing

(a) Describe “Call by Value Result”, “Call by Name”, “Call by Reference” parameter passing.

(b) Given the program:
```
program main (input, output)
    integer a, b;
    procedure p (x, y)
        begin
            x := x*a;
            y := y*z+x;
        end

    begin
        a := 2;
        b := 3;
        p (a, b);
        print a;
    end
```
What’s printed by the program assuming
6. Parsing

(a) Consider the following grammar:

\[
G ::= S \$
\]
\[
S ::= ( L )
\]
\[
| a
\]
\[
L ::= L , S
\]
\[
| S
\]

i. Convert the grammar to LL(1). Show all steps in order to get credit.
ii. Create the parse table and enter the proper production in each cell.
iii. Draw the FSA for the string (ab)+bbc.

(b) Consider the following grammar.

\[
S \rightarrow E \$
\]
\[
E \rightarrow E + T
\]
\[
| E - T
\]
\[
| T
\]
\[
T \rightarrow \text{num}
\]
\[
| \text{id}
\]

- Generate the Characteristic Finite State Machine (CFSM).
- Use your CFSM to parse the input x + 5 - 3. Use the table below. Note that you will not necessarily use up the entire table.
<table>
<thead>
<tr>
<th>Stack</th>
<th>Current token</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>