5 questions on 2 pages. 100 marks total. 50 minutes total
Open Book and Notes, Non-programmable calculators allowed, NO other electronic aids allowed
Answer all questions. WRITE LEGIBLY!
If you need to make any additional assumptions to answer a question, be sure to state those assumptions in your test booklet.
The line numbers on the left side of programs are for reference only and not part of the program.

1. [20 marks] Given the declarations in C:

```c
1 typedef struct { /* define dataStruct */
2     struct {
3         char name[5] ;
4         int key ;
5         double value ;
6     } data ;
7     unsigned char tag ;
8 } dataStruct ;
9
10 dataStruct A[ 100 ] ; /* array of dataStruct */
11 int i = 19 ;
```

Assume char is 8 bits aligned mod 8, int is 32 bits, aligned mod 32, double is 64 bits aligned mod 64.
Given the base address of the array A, show in detail the address calculation for the subscript reference
A[i + 7].data.value

2. [20 marks] Given an LL(1) parsing table constructed using the method discussed in lecture and a labelling
of the table rows (non terminal symbols) and the table columns (terminal symbols), give an algorithm to
reconstruct the Predict sets for a given non terminal symbol N.

3. [20 marks] Consider the following declarations in a Turing/Pascal-like language

```pascal
  type R : record
    ra : array 1 .. 100 of real
    rb : string (5)
    rc : record
      i, j : 10 .. 45 /* subrange type */
      rcb : boolean
    rd : int
  end record /* end R declaration */

  var X : R
  var Y : record
    ya : string(1)
    yb : R
  end record
```

Using a symbol and type table similar to the examples given in lecture, show the symbol and type tables that
would be created for these declarations.

Page 1 of 2 pages.
4. [15 marks] The Python programming language uses indentation rather than explicit `begin/end` or `{ }` characters to mark the beginning and end of blocks. This includes delimiting the bodies of functions and the bodies of control statements. For example:

<table>
<thead>
<tr>
<th>Python</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>def calc( x ):</code></td>
<td>define function calc</td>
</tr>
<tr>
<td><code>n = x * x + 7</code></td>
<td>assignment statement in calc</td>
</tr>
<tr>
<td><code>return n * n + 5</code></td>
<td>return statement in calc</td>
</tr>
<tr>
<td><code>def map( n, m )</code></td>
<td>define function map</td>
</tr>
<tr>
<td><code>if n &lt; m:</code></td>
<td>begin body of map</td>
</tr>
<tr>
<td><code>i = n - m</code></td>
<td>body of if statement</td>
</tr>
<tr>
<td><code>j = n + m</code></td>
<td>if statement continues</td>
</tr>
<tr>
<td><code>k = i * j</code></td>
<td>if statement continues</td>
</tr>
<tr>
<td><code>if n &gt; m:</code></td>
<td>start new if statement</td>
</tr>
<tr>
<td><code>i = n * m + 7</code></td>
<td>body of if statement</td>
</tr>
<tr>
<td><code>j = i * 2 + 5</code></td>
<td>if statement continues</td>
</tr>
<tr>
<td><code>k = i * j + 1</code></td>
<td>if statement continues</td>
</tr>
<tr>
<td><code>return k - 17</code></td>
<td>end if statement</td>
</tr>
<tr>
<td><code>print map( 17, 23 )</code></td>
<td>start of main program</td>
</tr>
</tbody>
</table>

Describe a method for scanning and parsing this language. In particular how would the scanner and parser interact to delimit blocks based on indentation?

5. [25 marks] Describe the semantic analysis checks that a Java compiler would perform on the following piece of Java code:

```java
class BreakDemo {
    public static void main(String[] args) {
        int[] arrayOfInts = {32, 87, 3, 589, 12, 1076, 2000, 8, 622, 127};
        int searchfor = 12;
        int i;
        boolean foundIt = false;
        for (i = 0; i < arrayOfInts.length; i++) {
            if (arrayOfInts[i] == searchfor) {
                foundIt = true;
                break;
            }
        }
        if (foundIt) {
            System.out.println("Found " + searchfor + " at index " + i);
        } else {
            System.out.println(searchfor + " not in the array");
        }
    }
}
```

Page 2 of 2 pages.