CSC 326, Final
December 15, 2004

Name: ____________________________
Student ID: ________________________

Signature: _________________________

• You have 120 minutes to write the 11 questions on this examination. A total of 125 marks is available.

• Justify all of your answers

• You may use one sheet of handwritten notes.

• Keep your answers short. If you run out of space for a question, you have written too much.

• The number in square brackets to the right of the question number indicates the number of marks allocated for that question. Use these to help you determine how much time you should spend on each question.

• Use the back of the page for rough work.

• Good luck

UNIVERSITY REGULATIONS:

• No candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.

• CAUTION: Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed form the examination and shall be liable to disciplinary action.

1. Making use of any books, papers or memoranda, electronic equipment, or other memory aid devices, other than those authorized by the examiners.

2. Speaking or communicating with other candidates.

3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

<table>
<thead>
<tr>
<th>Question</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/20</td>
</tr>
<tr>
<td>2</td>
<td>/14</td>
</tr>
<tr>
<td>3</td>
<td>/15</td>
</tr>
<tr>
<td>4</td>
<td>/10</td>
</tr>
<tr>
<td>5</td>
<td>/10</td>
</tr>
<tr>
<td>6</td>
<td>/9</td>
</tr>
<tr>
<td>7</td>
<td>/12</td>
</tr>
<tr>
<td>8</td>
<td>/10</td>
</tr>
<tr>
<td>9</td>
<td>/10</td>
</tr>
<tr>
<td>10</td>
<td>/8</td>
</tr>
<tr>
<td>11</td>
<td>/7</td>
</tr>
<tr>
<td>Total</td>
<td>/125</td>
</tr>
</tbody>
</table>
Question 1 [20 points]

• [3/20] What is aliasing?

• [2/20] Give at least one advantage and one disadvantage of dynamic type binding?

• [3/20] If a language does not have variable declarations (e.g., if C did not declare variables before they are used like this: int a;), how is type binding dealt with? (i.e., how does dynamic type binding work?)

(Question 1 continued on next page)
Miscellaneous

• [3/20] Explain the problem of type checking variables of union type.

• [3/20] What is the dangling pointer problem? Write a piece of code to illustrate it if you like.

• [3/20] Why is a reference type in C++ ’safer’ as an in/out parameter than pointers and call-by-value in C?

(Question 1 continued on next page)
Miscellaneous

• [3/20] If Espresso had out mode parameters, what would have to be changed (if anything) with respect to the way activation records are created and destroyed?

(End of question 1)
Languages

**Question 2** [14 points]

- [3/14] Is the following language regular \((n_1, n_2, n_3 \geq 0)\)?

\[ a^{n_1} b^{n_2} c^{n_2} \]

- [3/14] Consider the following grammar for a language \(L\):

\[ a(b|c)^*a^*(b^*|c) \]

Describe the strings created by this language.

- [8/14] Determine if the following strings are in the language \(L\) or not.

  - abaabb
  
  - abaabc

(Question 2 continued on next page)
Languages

- abcbbc
- acccbbb
Scoping

**Question 3** [15 points]


- [3/15] Consider the following program:

```c
int n;

int f(int n) {
    int a;
    a = 37;
    return 3*g(n+a);
}

int g(int b) {
    n=n+b;
    return n;
}

int h(int b) {
    int n = 2*b;
    return f(n+b);
}
```

Mark the static scope of the global variable \( n \) in the program above by underlining the appropriate lines of code.

(Question 3 continued on next page)
Scoping

• [3/15] Can you do the same for dynamic scoping? If so, do it; if not, explain why not.

• [3/15] For dynamic scoping, provide a call sequence where the $n$ in \texttt{return $n$} refers to the global $n$.

• [3/15] Also provide a call sequence where the $n$ in \texttt{return $n$} refers to the $n$ found in the function $f$.

(End of question 3)
Activation Records

Question 4 [10 points]
Consider the following main function along with the declarations of the functions $f$, $g$, and $h$ as well as the global variable $n$ from the previous question:

```c
void main() {
    n = 7;
    h(n);
}
```

Draw the activation stack, with detailed information for each activation record (with static and dynamic links etc.), as it would look immediately before the return statement in the function $g$ when executing the above main function.

(End of question 4)
Parsing

**Question 5** [10 points]
Consider the following context-free grammar:

\[
E ::= E + E \\
    | E \cdot E \\
    | \text{num}
\]

- [3/10] The above grammar is ambiguous with respect to operator precedence. Show a derivation of the string \(3 + 4 \cdot 7\) such that + takes precedence. Show the parse tree as well.

- [3/10] Now, show a derivation of the same string such that * takes precedence. Show the corresponding parse tree as well.

- [4/10] Rewrite the grammar such that the precedence rules are given by the grammar. That is, * takes precedence over +.

(End of question 5)
Short Circuit Evaluation

**Question 6** [9 points]

• [3/9] Briefly explain what short circuit evaluation is?

• [6/9] Consider the following piece of code:

```java
int a = 7;
int b = 3;
boolean c = (b<3) || (a++>18) && (b--<0);
```

Fill in the following table (the values for a, b and c after executing the 3 lines above):

<table>
<thead>
<tr>
<th>Value</th>
<th>With short circuit boolean evaluation</th>
<th>Without short circuit boolean evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(End of question 6)
Types and Function Overloading

**Question 7** [12 points]
Consider the following type lattice. You can imagine A, B, C, D, E and F as classes and the lines represent inheritance, for example: A is the super class of B, C is the super class of B and E, etc. (Note, in this language we allow multiple inheritance).

Given the following definitions:

```c
void f(A a, C c) { ... }
void f(B b, C c) { ... }
void f(B b, F f) { ... }
void f(D d, E e) { ... }
```

```c
da;
b;
c;
d;
e;
f;
```

for the following 4 invocations, determine which of the functions above get called, and explain why (You may refer to them as 1, 2, 3 and 4.):

(Question 7 continued on next page)
Types and Function Overloading

• [3/12] f(d,e)

• [3/12] f(b,c)

• [3/12] f(d,f)

• [3/12] f(d,f)

(End of question 7)
Parsing

**Question 8** [10 points]

Consider the following grammar:

\[
\begin{align*}
A & \rightarrow aB \mid aA \mid \epsilon \\
B & \rightarrow b \mid c \mid Ba
\end{align*}
\]

- [3/10] Is this grammar LL(1)? Explain why/why not.


- [4/10] Is the set of strings generated by starting at the symbol A a regular language? if 'no' explain why, if 'yes' write down the regular expression that represents the language.

(End of question 8)
First Sets

**Question 9** [10 points]
Consider the following grammar:

\[
\begin{align*}
T & \rightarrow bB \mid C \mid dB \mid \epsilon \\
B & \rightarrow b \mid C \\
C & \rightarrow c
\end{align*}
\]

- [6/10] Compute the the FIRST sets of \( T \), \( B \), and \( C \).

- [2/10] Is it LL(1)?

- [2/10] Is the grammar a regular language?

(End of question 9)
Question 10 [8 points]
What are the three major steps of the main execution loop of the EVM? Describe in a few sentences the workings of each of these steps?
Question 11 [7 points]
A binary palindrome is a string of bits (0 or 1) that reads the same from left to right as well as right to left.

Examples of binary palindromes include: 0, 1, 010, 11, 011001110.

Write a denotational semantics of a function called $P$ which determines if a string of binary digits is a palindrome. The function should take in binary strings and return a truth value.