CS 340 - Software Design
Midterm Exam #1 - Prof. Reed
Spring 2005

What is your name?:

There are two sections:
   I. True/False  40 points; (2 points each, 20 questions)
   II. Problems   60 points; (6 points each, 10 questions)

----------
100 points total

This test is worth 15% of your final grade. This test is open book and open notes. Use the back of the sheet if you need more room to write, although you shouldn’t need to. State your assumptions wherever it makes a difference. You have 60 minutes.

I. True False: (2 pts. each)

   T F  1. Although better designed, Netscape lost the browser war to Microsoft’s Internet Explorer (IE) due to marketing reasons.
   T F  2. Reverse engineering is a good example of Decomposition.
   T F  3. According to the class notes, the probability of a large software system project being cancelled is about 70%.
   T F  4. Both C++ and Java inherit language characteristics from Algol60.
   T F  5. Early FORTRAN programs used the first letter of a variable name to implicitly give the type for the variable.
   T F  6. Functional programming can be done not only in ML, but in Java as well.
   T F  7. Sometimes the use of a goto improves the legibility of a section of code.
   T F  8. In the classic article "GoTo Statement Considered Harmful," Dijkstra explains that our powers to visualize processes evolving in time are relatively poorly developed.
   T F  9. Modern high-level languages have constrained forms of goto.
   T F 10. Consider Pascal, with nested subprogram declarations. The static chain length and local offsets must be computed dynamically, since the calling sequence may vary.
   T F 11. Strings are immutable in Java.
   T F 12. When chaining constructors, a constructor call using this or super must be the first executable line in the method.
   T F 13. Every user-defined class automatically has toString() and equals() methods that can be used.
   T F 14. Assume class Collie inherits from Dog, which inherits from Animal. A Collie class method can not call an Animal constructor using super.super().
   T F 15. The access specifier of a method A() in a subclass that overrides the corresponding base class method A() can be no more restrictive than the access specifier of method A() in the base class.
16. A single *abstract* method makes an entire class *abstract*.

17. The Object class *clone()* method will work correctly if the object being cloned does not itself have any object references.

18. The type of an array in java can be a Class or an Interface.

19. Each object in Java (as compared to each class in Java) gets its own Dispatch Vector.

20. In Java an *Interface* can be used to share *global* variables between classes.

---

**II. Problems**: (6 pts. each) *Briefly* answer each of the following in the space provided.

1. What are the benefits of *abstraction* in designing and writing computer programs?

   *We can hide unnecessary detail and focus on what is important. We can do this for both data (as in UML diagrams) as well as for process (decomposition & step-wise refinement).*

2. Why would you want to pass a parameter by *constant reference*?

   *If it is very large, such as a large array. Passing as a value parameter would make a copy of it, which would be inefficient.*

3. Consider the relationship between *Exceptions* and *Assertions* in Java.

   a) In what type of situations would you want to use *Exceptions*?
   
   *Explicitly triggered errors, such as file not found.*

   b) In what type of situations would you want to use *Assertions*?

   *Expensive internal errors, such as pre- & post condition checking for a method. Assertions can be disabled or enabled selectively at run time, which is why it is useful to call computationally expensive code with assertions.*
4. Consider the PROLOG predicates shown at left below.

parent( pam, bob).
parent( tom, bob).
parent( tom, liz).
parent( bob, ann).
parent( bob, pat).
parent( pat, jim).
female( pam).
male( tom).
male( bob).
female( liz).
female( ann).
female( pat).
male( jim).

offspring( Y, X) :-
    parent( X, Y).
mother( X, Y) :-
    parent( X, Y),
    female( X).

grandparent( X, Z) :-
    parent( X, Y),
    parent( Y, Z).
sister( X, Y) :-
    parent( Z, X),
    parent( Z, Y),
    female( X),
    different( X, Y).

predecessor( X, Z) :-
    parent( X, Z).
predecessor( X, Z) :-
    parent( X, Y),
    predecessor( Y, Z).

What is the output of
Circle your answer.

?- grandparent( bob, jim).

5. Why would a programmer need to be aware that there is no short-circuit-evaluation in PASCAL?

It means the following code that searches for a key value in an array
would give a runtime error:

\[
\text{index} = 0; \\
\text{while}(\text{index} < \text{limit}) \land \text{key} \neq \text{array[index]} \text{ do} \text{index} = \text{index} + 1; \\
\]

When this is no longer true, with the index past
the end of the array, this section is still
executed, which generates an error.
6. Assume you are given the following program:

```plaintext
PROGRAM main;

VAR x: INTEGER;

PROCEDURE sub1; BEGIN { sub1 }
   WRITELN('x = ', x)
END; { sub1 }

PROCEDURE sub2; VAR x: INTEGER;

PROCEDURE sub3(INTEGER x); BEGIN { sub3 }
   x := 3;
   sub1;
END; { sub3 }

BEGIN { sub2 }
   x := 2;
   sub3(x);
END; { sub2 }

BEGIN { main }
   x := 1;
   sub2;
END. { main }
```

What value of `x` is printed by the `WRITELN` in `sub1` if dynamic scoping is used? Circle your answer.

Dynamic scope resolution follows the calling sequence. `sub1` was called from `sub3` (its dynamic parent), so the nonlocal value of `x` is the one found in `sub3` which has a value of \( \Box 3 \)

7. Consider the relationship between an Interface and an Adapter Class. Give a conceptual example illustrating the advantage of using an Adapter Class.

One Interface is that for using a mouse. It has multiple methods declared. To implement this Interface would require giving the definition for each of these methods. It is more convenient to first create an Adapter class that itself gives the definition for each method. Then we can subclass the Adapter class, overriding only the method(s) we need.
8. Consider the program outline shown at left below. Using the blocks shown at right below, draw the
stack using a single block for each stack frame corresponding to the calling sequence:
main -> bigsub -> sub1 -> sub2 -> sub4 -> bigsub -> sub2
For each stack frame show only the static links. Assume the call to main uses the stack.

9. Consider the following code:

```c
int value;  /* Global variable */
int list[4];  /* Global array of two members, subscripted by 0 & 1 */
void swap (int *a, int *b)
{
    int temp = *a;
    *a = *b;  /* temp = b */
    *b = temp;  /* temp = a */
}

void main()
{
    value = 3; list[0] = 1; list[1] = 0; list[2] = 3; list[3] = 2;
    swap( list[0], list[1]);
    swap( value, list[value]);
}
```

What are the values of the variable "value" and the array "list" at the end of the above program if parameters are passed by name?:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
10. Draw the UML diagram corresponding to the following:

Class **Teacher**, with private instance variable `ssn` as an int, and private instance variable `name` as a String. Provide appropriate accessor and mutator methods.

Class **KindergartenTeacher** (that inherits from class Teacher), with public instance variable `specialSkill` as a String. Class KindergartenTeacher should also make implement an Interface called **Ordered**.