CPTS 355 Sample Final Exam

NAME _____

Directions: This exam has 11 pages (including this one) printed double-sided. None should be blank. Check now to make sure your exam is complete. Answer/solve all the questions/problems. If a question asks "why", asks you to "explain", or asks you to "describe" then you must write in complete sentences.

The test is open book and open lecture notes and other reference materials. You may **not** use a computer, PDA, cell phone, etc. during the test.

1a) [5] What value is left on the top of the stack after executing this PostScript program? Explain how you determined your answer.

```
/a 4 def
/b {/a 6 def} def
b
a
```

1b) [5] What is the value of the following ML expression? Explain how you determined your answer.

```
let
   val a = 4
   fun b () = let
      val a = 6
   in
      5
   end
in
   b(); a
end
```

1c) [5] What is the value of the following ML expression? Explain how you determined your answer.

```
let
   val a = ref 4
   fun b () = a := 5
in
   b(); !a
end
```

(2) [5] Recall that in Scheme, DEFINE is a special form. An example of the use of DEFINE is `(DEFINE x 6)`. Why is it necessary that DEFINE be a special form and not a function?

3) [15] Identify each of the following as an example of parametric polymorphism, inheritance polymorphism or ad-hoc polymorphism.

a) (ML)

```
fun len ([]: 'a list) = 0
  | len (_::xs) = 1 + len(xs)
```

b) (Java)

```
class A {
     int x;
     public A() {x = 0}
     public A(int y)  { x = y}
}
```

c) (Java)

```
class A {
   public int x;
}
class B extends A {
   public int y;
}
A a = new B();
```

4) [10] Consider the following ML code fragment::

```
exception oops of int (* means oops carries an int value *)
fun p (x: int) =
   if x=0 then raise oops(7)
   else 5
fun q () =
   p(0) handle oops(x) => x-3
q()
```

What value is returned by `q()`? Why?

5) [15] (**three parts**) Multiple inheritance poses a number of problems for language designers and implementers and different languages make different choices about how to solve these problems. In the following code identify instances of

a) a name clash

b) diamond inheritance

c) and name a class for which two different vtables would have to be provided.

```
class A {
   public:
      int x;
      virtual void f();
}
class B: public A {
   public:
      virtual void g();
}
class C: public A {
   public:
      virtual void f();
}
class D: public B, public C {
   public:
      int y;
      virtual void g();
}
A a; B b; C c; D d;

d.g(); d.f(); c.f(); b = d;
```

6) [20] Draw the call stack at the point marked by ← in the following Python program. Clearly label each activation record with the name of the procedure that it represents and show the static and dynamic links in each activation record. (Recall that in Python assigning to a variable creates an instance of that variable in the scope where the assignment occurs; so in this program there are separate variables named y associated with procedures A and B.)

```
def main ():
   x = 2
   def A ():
      y = 4
      def B ():
         y = 8
         C()
      def C ():
         z = y+x ←
         print z
      B()
   A()
```

Using your diagram of the call stack, explain how the uses of variables x and y at the ← are resolved to the correct binding occurrences (either the one in A, the one in B, or the one in Main).

7) Consider the following code in the language C-by-ref which is like C but parameters are passed by reference.

```
int x = 4
void p(int y, int z) {
   y = y+2;
   z = z*3;
}
p(x,x);
```

7a) [5] What is the value of variable x at the end of execution? Why?

7b) [5] If the parameter passing rule was pass-by-value-result in the above code, what value would x have at the end of execution? Why? Explain why there may not be a unique answer to this question.

8) [5] Problem 13.2 in the textbook asked you to describe uses for the Java keywords “final” and “finalize”. “finally” is also a Java keyword. It is used as part of a try-except statement (p. 399). Unfortunately, the description of the finally block in the book is incorrect – or at least incomplete. It says “If an exception is raised and caught, then the sequence of statements following the keyword “finally” will be executed after the exception handler has finished.” In fact, the code of a “finally” block is executed regardless of how the try block terminates, whether by an exception handled locally, an exception handled higher up the stack, or by normal exit. Argue that the correct definition is the more useful one. If you can, show an example of how a “finally” block would be helpful.

Refer to the following code for problem 10.

```
typedef struct Trouble {
   int i;
   struct Trouble *t;
} *TroubleP;
while (1) {
   TroubleP p = (TroubleP) malloc(sizeof *p);
   p->t = p;
}
```

9) [5] Suppose that C had automatic memory reclamation using reference counting. What would be the result of executing the above code? Why?

10) [5] How would using a mark-and-sweep collector change the behavior of the program in problem 9?

11) [5] What is the primary advantage of a copying garbage collector over a mark-and-sweep collector? Why?

12) Let x$y stand for the operation (xy/(2x+y)).

12a) [2] What is the value of 2$2$2 if $ associates to the right?

12b) [2] What is the value of 2$2$2 if $ associates to the left?

13) [10] Consider the CFG:

```
<e> ::= <e> $ <f>
          | <f>
<f> ::= <b>^<f>
          | <b>
<b> ::=  y | z | w
```

The expression `y^z $ y^z^w` can be derived using this grammar as follows:

```
<e> => <e> $ <f>
⇨  <f> $ <f>
⇨  <b>^<f> $ <f>
⇨  <b>^<b> $ <f>
⇨  y^z $ <f>
⇨  y^z $ <b>^<f>
⇨  y^z $ y^<b>^<f>
⇨  y^z $ y^z^<b>
⇨  y^z $ y^z^w
```

Draw the parse tree corresponding to this derivation.

14) [10] In C++, some objects are allocated in the heap while others are allocated on the stack. Consider the following declarations:

```
class A {public: int x;};
class B: public A {public: int y;};
A a;
B b;
A *pa = new A;
B *pb = new B;
```

A potentially confusing aspect of C++ programs is that the behaviors of the assignments `a=b` and `pa=pb` are subtly different. Explain why.

In Java this confusion does not occur. Why not?

15) [10] A test-and-set instruction has the effect of executing the following procedure atomically, that is, indivisibly.

```
int testAndSet (int *x) {
   int result = *x;
   *x = 1;
   return result;
}
```

Fill in the blanks in the following code fragment using testAndSet() to ensure that two processes do not both enter the same section of code at the same time (mutual exclusion).

```
int lock = 0;
…
while(_________________) {
   ____________________________
}
… code to be executed with mutual exclusion…
/* allow another process to enter */
_____________________________
```

16) [15] Write a recursive ML function that computes the same value as this Python function. Hint: you may want to use an auxiliary function. (Note here that `l` is a list of integers, `x` and `s` are integers.)

```
def f (x, l, s):
   sign = 1
   for a in l:
      s = s*x + a*sign
      sign = -sign
   return s
```

17a) [10] For the program

```
i := 1
j := 1
sum := 0
while (i<n) {
   sum := sum + j
   j := j*2
   i := i + 1
}
```

which of the following is a loop invariant for the `while` loop. You *must* show work establishing that your choice indeed is an invariant in order to receive credit (i.e., unsupported guesses won't count for much).

a) i<n and sum = $(\sum_{k=1}^{i-1} 2^k) - 1$ and $j = 2^{i-1}$

b) i>=n and sum = $\sum_{k=1}^{i} 2^{k-1}$ and $j = 2^{k-1}$

c) i<=n and sum = $\sum_{k=1}^{i-1} 2^{k-1} - 1$ and $j = 2^{k-1}$

d) i<=n and sum = $\sum_{k=1}^{i-1} 2^{k-1}$ and $j = 2^{i-1}$

17b) [5] Based on the invariant you chose above and the loop inference rule, what is the value of sum when the loop terminates?