

Tempo a disposizione: ore 2.

1. Si descrivano brevemente vantaggi e svantaggi di due metodi per la gestione a run-time dello scope statico.
2. Si dica se la seguente grammatica è ambigua oppure no, dimostrando quanto asserito:

$$(\{S, C, B\}, \{\text{if, then, else, skip, true, false}\}, \{S\}, R)$$

dove R è il seguente insieme di regole:

$$\begin{aligned} S &\rightarrow C; \\ C &\rightarrow \text{skip} \mid \text{if } B \text{ then } C \text{ else } C \mid \text{if } B \text{ then } C \\ B &\rightarrow \text{true} \mid \text{false} \end{aligned}$$

3. Si assuma di avere uno pseudolinguaggio che adotti la tecnica del *reference count*; se OGG è un generico oggetto nello heap, indichiamo con $OGG.ref-c$ il suo reference count (nascosto). Si consideri il seguente frammento di codice:

```
class C { int n; C next;}
C foo = new C(); // oggetto OG1
C bar = new C(); // oggetto OG2
foo.next = bar;
bar.next = foo;
foo = new C(); // oggetto OG3
bar = foo;
```

Si dia il valore di $OG1.ref-c$, $OG2.ref-c$ e $OG3.ref-c$ dopo l'esecuzione del frammento. Quali di questi tre oggetti possono essere ritornati alla lista libera?

4. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping statico e passaggio di parametri per valore e per nome.

```
{int x = 0;

void pippo(value int y, name int z){
    z = x + y + z;
}

{ int x = 1;
  int y = 10;
  int z = 20;
  pippo(x++, x);
  pippo(x++, x);
  write(x);
}

write(x); }
```

(la primitiva `write(x)` permette di stampare un valore intero; un comando della forma `foo(w++)`; passa a `foo` il valore corrente di `w` e poi incrementa `w` di uno).

5. Si scriva, in un qualsiasi (pseudo-)linguaggio, un frammento di codice tale che il numero dei record di attivazione (RdA) presenti a run-time sulla pila fra il RdA di una fissata chiamata di procedura e quello del blocco che la contiene, dipenda dal valore di una variabile letta dall'esterno.

6. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che permette il passaggio di funzioni come parametro, usa deep binding, scoping statico e passaggio per valore.

```
{int x = 0;
 int y = 0;
 int somma(value int z){
     return x+y+z;
 }

 int f( int function g (value int n)) {
     int x = 10;
     int y = 10;
     return g(x)+g(y);
 }

 { int x = 50;
  int y = 50;
  x= f(somma);
  write(x);
 }
}
```

7. Si dica cosa viene stampato, dal frammento di codice del precedente esercizio nel caso in cui si usi deep binding e scope dinamico.
8. **Solo per: corso AL; corso MZ a.a. 2002/03** Supponiamo di rappresentare i naturali usando 0 per lo zero e $s(N)$ per il successore di N (quindi i naturali sono $0, s(0), s(s(0)) \dots$). Sia t un termine che rappresenta un numero naturale (come detto sopra) e X una variabile. Si dica qual è il termine associato alla variabile X al termine della valutazione del goal $p(t,X)$ nel seguente programma

```
p(0,0).
p(s(Y), s(s(Z)) ):- p(Y,Z).
```

9. **Solo per il corso MZ a.a. 2004/05:** Si descriva sinteticamente la nozione di genericità. Si dia poi un esempio di classe generica in Java2SE 5.0 (Sett 2004).