

Comparative Programming Languages

UCT CSC304 – Class Test – 2003

1. “If Prolog was object-oriented, it would be a universal programming language”. Discuss two arguments to dispute this statement. [4] (universal means that we do not need anything else)

you cannot write an operating system in Prolog.

Prolog is not efficient for some classes of high-performance problems.

2. Besides Pascal’s variant records and C’s unions, list two other examples of how aliasing can occur in a programming language (such as C++). [2]

call-by-reference parameters

pointers

3. What do dynamic typing and dynamic scoping of variables have in common (besides the word “dynamic”)? [2]

both refer to attributes of variables that are only bound at runtime.

4. Consider the following program in an ALGOL-like statically-scoped language, using displays.

```
program main ()
  subprogram funca ()
  {
    funcb ();
    // breakpointX
  }
  subprogram funcb ()
  {
    subprogram funcce ( int x )
    {
      x = 0;
      // breakpointY
    }
    funcce (6);
  }
  // breakpointZ
  funca ();
}
```

4.1. What would the display look like at each of the 3 breakpoints indicated? [6]

4.2. Can a display be used for a dynamically-scoped language? If so, how – if not, why? [3]

```
X:   1 --->   funca
      0 --->   main
Y:   2 --->   funcce
      1 --->   funcb
           funca
```

```
      0 ---->   main
Z:    0 ---->   main
```

No. The entire call sequence must be searched for non-local references and displays override/hide subprograms at the same level.

5. In languages that allow side-effects, short-circuit evaluation of Boolean expressions can result in incorrect or unpredictable results. Explain how this can occur. [2]

if evaluation of some terms are not needed to decide a boolean expression, their potential side-effects (e.g., by subprogram calls) do not occur, changing the semantics of the program.

6. Most languages that support exceptions choose not to return control to the statement that raised the exception. Why? [2]

the exception could occur again if the cause was not removed, resulting in an infinite loop.

7. In some languages semaphores are part of the basic syntax but in others the functionality is provided through a library. State two possible advantages of the latter approach. [4]

if that functionality is not used, it is not part of the compilation/linking process.

the basic syntax is simplified by not including features that can be implemented through libraries.