

# Comparative Programming Languages

## UCT CSC304 – Class Test Supp – 2003

1. “C++ is a universal programming language”. Discuss two arguments to support this statement. [4] (universal means that we do not need anything else)

you can write everything from an OS to a GUI application.  
it is extensible and libraries are available for most applications.

2. Why can we not write regular expressions to match arbitrary fragments of Java code? [2]

regular languages do not have recursion – and most block structured languages, like Java, are defined recursively.

3. How is the lifetime of variables affected by the choice between static and dynamic scoping? [2]

it isn't – these are relatively independent attributes of variables.

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

```
program main ()
  subprogram funca ( int x )
  {
    subprogram funcb ()
    {
      subprogram funcce ( int x )
      {
        // breakpointX
      }
      funcce (6);
    }
    // breakpointY
    funcb ();
  }
  funca (12);
  // breakpointZ
}
```

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

4.2. Compared to static chains, displays provide faster access to non-local references at the expense of both speed and auxiliary storage. Explain how speed and storage are affected. [4]

```
X:   3 ---->   funcce
     2 ---->   funcb
     1 ---->   funca
     0 ---->   main
Y:   1 ---->   funca
     0 ---->   main
Z:   0 ---->   main
```

displays require time to restore the display after each subprogram returns while static chains do not.

additional (auxiliary) space is required for the display itself.

5. What is the value of the variable “c” after execution of the code below if the parameter is a) pass-by-value b) pass-by-reference c) pass-by-name? [3]

```
b = 6;  
func xyz ( integer a ) returns integer  
{  
    a = 3;  
    b = 4;  
    return a+b;  
}  
c = xyz (b);
```

a) 7    b) 8    c) 8

6. What is the effect of aliasing on statement-level concurrency (as appeared in ALGOL68)? [2]

without mutual exclusion, two concurrent statements could modify the same variable at the same time, resulting in unpredictable/incorrect results.

7. Monitors are not a feature of all programming languages. How is access to shared data mediated in languages without monitors? [2]

using lower-level constructs such as semaphores.