

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)
2. Why can we not write regular expressions to match arbitrary fragments of Java code? [2]
3. How is the lifetime of variables affected by the choice between static and dynamic scoping? [2]
4. Consider the following program in an ALGOL-like statically-scoped language, using displays.

```
program main ()
  subprogram funca ( int x )
  {
    subprogram funcb ()
    {
      subprogram funcc ( int x )
      {
        // breakpointX
      }
      funcc (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]
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);
```

6. What is the effect of aliasing on statement-level concurrency (as appeared in ALGOL68)? [2]
7. Monitors are not a feature of all programming languages. How is access to shared data mediated in languages without monitors? [2]

