Comparative Programming Languages

UCT CSC304 – Final Exam – November 2003

Question 1 is compulsory. Then answer either question 2 OR 3.

Question 1 – General Concepts [15]

1.1. "HTML is a programming language". State whether or not you agree, and provide a suitable reason. [2]

No. HTML is used to specify the structure of documents and not algorithms.

1.2. C++ is not an orthogonal language. Discuss one feature, or lack of a feature, to justify this statement. [2]

Arrays cannot be added together using the "+" operator that applies to integers.

1.3. Draw the stack of activation records corresponding to the following ALGOL-like program when it is at "breakpointX". [5] (Assume static chains and include all parameters).

```
program main ()
   subprogram funca ()
   {
      funcb ();
   }
   subprogram funcb ()
   {
      subprogram funcc ( int x )
      {
         x = x + 1;
      }
      funcc (6);
      // breakpointX
   }
   funca ();
}
```

```
funcb   static link    -----+
        dynamic link   --+  |
        return (funca)   |  |
                       <-+  |
funca   static link    -----+
        dynamic link   --+  |
        return (main)    |  |
                       <-+  |
main                        |
                       <----+
```

1.4. 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-value-result c) pass-by-reference? [6]

```
c = 12;
subprogram xyz ( integer a )
{
   a = a + c;
   c = c + a;
}
xyz (c);
```

a) 36 b) 24 c) 48

Question 2 – Subprograms and Scope [10]

2.1. How do subprogram side-effects affect the maintainability of programs in a language? [2]

It is more difficult to maintain programs where the effect of subprograms is not self-contained, thus making if difficult to understand the subprograms in isolation.

2.2. In a statically-scoped language, why are dynamic links still required in activation records? [2]

To return the stack to its state before each subprogram call.

2.3.1. If an ALGOL-like language did not allow re-entrant subprograms (e.g., by recursion), are static links still required in stack-based activation records? Give a brief reason for your answer. [1]
2.3.2. Are dynamic links still required? Give a brief reason for your answer. [1]

2.3.1. yes – because it is still possible to have non-local references to other subprograms.
2.3.2. yes – because they are needed to restore the stack after subprogram returns.

2.4. Both templates and object-oriented polymorphism are mechanisms that support generic subprograms. Discuss one advantage and one disadvantage of the template approach, when compared to the object-oriented polymorphism approach. [4]

adv: templates have faster execution times
disad: templates result in larger code size

Question 3 – Higher-level Languages [10]

3.1. Perl adopts the "kitchen sink" approach to languages by incorporating multiple redundant features from different paradigms. Explain how this affects readability and writability of programs in Perl. [4]

decreases readability since there are a lot of different constructs to learn and understand.
increases writability since there are many different ways to implement an algorithm.

3.2. Perl programs cannot be fully compiled (like C++). Explain why. [2]

some features require the full interpreter to be present e.g. “eval”

3.3. In Borland Delphi, an image component can display a preview of its contents while the program is being written (i.e., before the program using it has been compiled). What features of the programming language and design environment enable this functionality? [4]

The language has support for distinct design-time and runtime methods to display components. At design-time, the design-time methods are used by the IDE to update the display and provide previews of the component. At runtime, the runtime methods are used to enable the actual functionality.