

CSC 533: Programming Languages

Sample Midterm Exam Questions

1. True or False?

- _____ A *strongly-typed* language is one in which all variables are bound to a type at compile time and those type bindings cannot be changed.
- _____ Languages that statically bind variables to types and addresses tend to be *compiled* rather than interpreted.
- _____ In both C++ and Java, the actual *sizes* of the primitive types (e.g., int, float, double) may differ from implementation to implementation.

2. Short Answer

A. Even though Java has only one type of parameter passing, pass-by-value, it appears that objects are treated differently than primitive values when passed as parameters. For example, when you pass an int value as a parameter to a method, any changes to the corresponding variable within the method do not affect the variable in the call. However, if you pass an ArrayList as parameter, you can add or remove from the corresponding list within the method, and those changes affect the ArrayList in the method call. Explain this apparent inconsistency.

B. Consider the following BNF grammar rules that define expressions involving addition, subtraction, and sign prefixes.

```
<expr> → <pre> '+' <expr> | <pre> '-' <expr> | <pre>  
<pre> → '+' <term> | '-' <term> | <term>  
<term> → '(' <expr> ') ' | <id>  
<id> → 'A' | 'B' | 'C' | 'D'
```

What is the associativity of the subtraction operator? In particular, how would the expression A-B-C be parsed? Justify your answer.

3. Specific Topics

Consider the ALGOL-like program on the right:

A. In this program, the subroutine OUTPUT is called twice. Draw the contents of the run-time stack *at each point in the execution after OUTPUT is called*. For each activation record on the stack, be sure to indicate both the static and dynamic links for that record, as well as the values of any parameters and local variables. Assume that parameters are passed by-value.

1st call

2nd call

```
program MAIN;
  int J, K;

  procedure OUTPUT();
  begin
    print J, K
  end;

  procedure ONE(int K);
  begin
    if (K < 0)
      ONE(-K);
    else
      OUTPUT();
  end;

  procedure TWO();
  int J;
  begin
    J := 55;
    ONE(100);
  end;

begin
  J := 9;
  K := 23;
  ONE(-14);
  TWO();
end.
```

B. What is output by this program if *static scoping* is used?

C. What is output by this program if *dynamic scoping* is used?