

# Final Exam

Programming Language Concepts

Due December 13, 2020

1. (5 points) For what types of A and B is the simple assignment statement  $A = B$  legal in C++ but not Java?
2. (5 points) What are the benefits and drawbacks of replacing of all implicit type conversion with explicit type conversion in a programming language? Logically support your argument with examples. Discuss the benefits and drawbacks using the evaluation criteria of programming languages discussed in this class.
3. (18 points) The following problem list the operators line by line, where each line is a higher precedence than the operators underneath.

```
( postfix ) ++, ( postfix ) --
++ ( prefix ), -- ( prefix )
*, >= , &
+ , - , <=
- ( unary ) , + ( unary ) , %
> , <
&&, /, !
||, ~| ( this symbol meant to be a tilde followed by a pipe )
=, /=
```

This problem will be using a left associative property on every operand that defaults to always doing the highest level operation from left to right.

Show the order of evaluation of the following expressions by parenthesizing all sub-expressions and placing a superscript on the right parenthesis to indicate order. Like in the example from the chapter 7 problem set.

- 1)  $a * b - 1 + c$
- 2)  $a * (b - 1) / c \% d$
- 3)  $(a - b) / c \& (d * e / a - 3)$
- 4)  $(a + b \leq c) * (d > b - e)$
- 5)  $-a \ || \ c = d \ \&\& \ e$
- 6)  $a > b \ \sim| \ c \ || \ d \leq 17$
- 7)  $-a + b$
- 8)  $a + b * c + d$
- 9)  $E = ++(a++)$

4. (9 points) Solve the problems for the above expressions ( from problem 3 ) and show your work for the values ( assume 5 bits are used, two's compliment notation,  $\sim|$  and  $\&$  represents the logical bitwise XOR and AND operations respectively,  $!$  Represents boolean NOT , and there is implicit type conversion between BOOLEANS and INTEGERS )

A = 5

B = 7

C = 11

D = -13

E = -2

5. (9 points) Write a formally defined CFG for the above problem ( from problem 3 ) and assume and imply that:
  - 1) only variables can have the increment or decrement operation;
  - 2)  $\sim|$  and  $\&$  represents the logical bitwise XOR and AND operations respectively,
  - 3)  $!$  Represents boolean NOT

6. (18 points) Rewrite each expression from problem 3 as function calls as if it were an object oriented programming language ( like how is practically done in Ruby )? Is there a need to express this using a symbol to break precedence? Why or why not?
7. (11 points) Write an RDA for the above mentioned problem 6?
8. (15 points) Write code to describe how you would analyze the semantic meaning of the languages of the expressions high lighted above? How would it track variable value or type? Does value or type matter? If the do matter what are your rules around the operations that they matter for?
9. (5 points) Evaluate  $a > b > c$  in math logic. Evaluate the same inequality in C. Explain the difference between the two.
10. (5 points) Let the function fun be defined as

```
int fun(int *k) {  
    *k += 4;  
    return 3 * (*k) - 1;  
}  
  
int main() {  
    int i = 10, j = 10, sum1, sum2;  
    sum1 = (i / j) + fun(&j);  
    sum2 = fun(&i) + (i / j);  
    return 0;  
}
```

Run the code in on some system that supports C and edit it to determine the values of sum1 and sum2. Explain the results