

# COSC4315 Midterm Exam

## Multiple Choice #1

### 1. Instructions

The exam is individual. You can use your computer and any Internet resource. You cannot ask TAs or professor any clarification. Each question is worth 10 points. Choose one answer. Write one question # and one chosen letter per line in your answer file.

### 2. Questions

1. Which is the regular expression for an identifier in Python? Ignore the underscore.

- (A) `letter(letter|digit)*`
- (B) `letter(letter|digit)+`
- (C) `letter+(letter|digit)*`
- (D) `letter*(letter|digit)+`
- (E) `letter*(letter|digit)*`

variable names  $[(A-Z), (a-z), (-), (0-9)]$

"|"  $\Rightarrow$  union  
 $\{ "ab", "c" \} + \{ "d", "e" \}$   
 $\Rightarrow \{ "ab", "c", "d", "e" \}$

\* Zero or more occurrences —  
 $ab^*c \Rightarrow 'ab', 'abc', 'abbc'$

+ one or more occurrence  
 $ab^+c \Rightarrow 'abc', 'abbc', 'abbbc'$

$\rightarrow$  modify the original value.

2. Which features in Java can produce variable mutation?

- (A) = var. assignment, ++ increment, while loop
- (B) ~~function definition~~, ++ increment, while loop
- (C) ~~function definition~~, recursive function, while loop
- (D) = var. assignment, pass parameter by value, for loop
- (E) ~~function definition~~, pass parameter by reference, for loop

3. Consider if statements in Python. Which one is false?

- (A) the if statement has an ambiguous grammar rule
- (B) if statements can be nested multiple times, inside if/elif/else ✓
- (C) the elif clause cannot be expressed with if/else
- (D) the if/else statement by itself cannot produce mutation ✓
- (E) The else/elif parts are optional ✓

4. Consider the functional expression: `multiply(add(2,3),add(add(7,9),5))`. which is the correct postfix expression?

- (A) ~~2 3 add 7 add 9 5 add multiply~~
- (B) ~~2 3 7 9 add add add 5 multiply~~
- (C) ~~2 3 add 7 9 add add 5 multiply~~
- (D) 2 3 add 7 9 add 5 add multiply
- (E) 2 3 add 7 9 5 add add multiply

$2\ 3 +\ 7\ 9 +\ 5 +\ *$

<https://www.mathblog.dk/tools/infix-postfix-converter/>

5. Consider a function definition in Python based on the functional if operator (similar to C++ ternary operator ?). Which is the correct recursive statement to sum all numbers in a list? Example `sum_if(L=[3,5,7])`.

- (A) `return l[0] if len(l)==0 else l[0]+sum_if(l[0:])`
- (B) `return l[0] if len(l)==0 else l[0]+sum_if(l[1:])`
- (C) `return l[0] if len(l)==1 else l[0]+sum_if(l[0:])`
- (D) `return l[0] if len(l)==0 else l[0]+sum_if(l[1:])`
- (E) `return l[0] if len(l)==1 else l[0]+sum_if(l[1:])`

6. Which data structure is bad to store variables (data type, value) in a programming language?

- (A) Hash table ✓
- (B) Sorted array

must all be of same type  
`int arr[3] = { "str" }  $\rightarrow$  invalid.`

- (C) Binary search tree ✓
- (D) Linked list ✓
- (E) None of the above

Strongly Typed  $\Rightarrow$  type errors are always detected

7. Compare C, C++ and Java. Which one is true?

- (A) C is strongly typed, C++ is strongly typed, Java is strongly typed
- (B) C is weakly typed, C++ is strongly typed, Java is strongly typed
- (C) C is strongly typed, C++ is weakly typed, Java is strongly typed
- ~~(D) C is weakly typed, C++ is strongly typed, Java is weakly typed~~
- ~~(E) C is weakly typed, C++ is weakly typed, Java is weakly typed~~

8. Which of following  $\lambda$  calculus expressions is valid?

- ~~(A)  $(\lambda x.yx)\lambda x$~~
- ~~(B)  $(\lambda yx)\lambda y.y$~~
- ~~(C)  $(\lambda x.yx)\lambda y$~~
- ~~(D)  $(\lambda yx)\lambda y.y$~~
- ~~(E)  $(\lambda x.yx)\lambda y.y$~~

<https://projectultimatum.org/cgi-bin/lambda>

9. Which C++ features were removed in Java?

- I. functions not contained in a class
- ✓ II. pointers and memory deallocation operators
- ✓ III. Recursive functions which can produce variable mutation

- ~~(A) I~~
- ~~(B) II~~
- ~~(C) I, II~~
- ~~(D) I, III~~
- (E) II, III

non-value normal forms are called stuck

10. Which of the following expressions is not stuck in  $\lambda$  calculus?

- (A)  $\text{pred}(0)$  ✓ slide 422
- (B)  $\text{iszero}(\text{false})$  ✓ slide 464
- (C)  $\text{succ}(\text{false})$
- (D)  $\text{iszero}(\text{succ}(\text{true}))$
- (E)  $\text{succ}(\text{true})$  ✓ slide 464

slide 405

### An abstract syntax

```
t ::=
  true
  false
  if t then t else t
  0
  succ t
  pred t
  iszero t
```

Terms defined by a BNF style grammar.  
Not worried about ambiguity.  
 $t$  is a syntactic metavariable