CS 3360, Spring 2019
Professor Ward

____

Final Examination

/102

165 minutes. Closed book. 3 pages of handwritten notes allowed. Answer concisely.
If you leave the classroom for any reason, your test will be graded on only what you did up until that time. This also applies to use of the restroom.

1. [4] In each of the following pairs of properties, circle the one that has become more prevalent in more modern programming languages.

   a. compiled *vs.* interpreted
   b. more readable *vs.* more efficient
   c. has pointers *vs.* has reference types
   d. implicit memory deallocation *vs.* explicit memory deallocation

2. [5] For each of the following errors, describe its likely source, for example, which stage of the compilation process. If there are two possibilities, list both.

   a. array index out of bounds

   b. type mismatch

   c. missing semicolon

   d. unexpected character א found in program source

   e. no definition or declaration found for identifier wordCount

3. [8] The book distinguishes between data abstractions and control abstractions. Mark the following with D if it's clearly a data abstraction, C if it's clearly a control abstraction, and B if it has elements of both. Optionally explain some of your answers if you wish.
   a. ___ classes
   b. ___ interfaces
   c. ___ functions
   d. ___ procedures
   e. ___ list comprehensions in Haskell
   f. ___ rule statements in Prolog

/17

4. [5] When writing a compiler, for which five of the following purposes might you use an attribute grammar?
    a. detecting misspelled reserved words
    b. determining when to insert instructions to convert an integer to a float
    c. detecting type mismatch errors in expressions
    d. detecting type mismatch errors in assignment statements
    e. detecting type mismatch errors in function calls
    f. detecting if the same variable is declared in two places
    g. detecting infinite loops
    h. handling runtime exceptions

5. [5] We are interested in the language consisting of the set of all strings which are true English statements about elephants. The tokens of the language are *fly, die, are, elephants, big, small, tiny, miniscule, eat* and *!*. Sentences in the language include {*elephants die, elephants eat, elephants are big, elephants are big big big!*}. Sentences not in the language include {*elephants are tiny, elephants fly, tiny big elephants fly, tiny tiny, are die big tiny*}
    a. Write a context free grammar for this language using BNF or EBNF. Document any assumptions you make.
    b. Was your grammar ambiguous?

6. [2] (from json.org) In JSON, An object is an unordered set of name/value pairs. An object begins with {$_{\text{left brace}}$ and ends with }$_{\text{right brace}}$. Each name is followed by :$_{\text{colon}}$ and the name/value pairs are separated by ,$_{\text{comma}}$. A value can be a string in double quotes, or a number, or **true** or **false** or **null**, or an object. These structures can be nested.

    Correct all errors, if any, in the grammar for objects below.

    Object → { } | {Members}

    Members → Member | Member, Members

    Member -> String : Element

    Element -> Value

    Value -> Object | String | Number | **true** | **false** | **null**

/12

7. [3] Algol is statically typed, uses infix notation, and has no overloaded operators. In the Algol statement: **count := count + 5;**
    a. the type of **count** is bound at _______
    b. the meaning of the operator **+** is bound at _________
    c. the value of **count** is bound at _________

    a. compile time
    b. run time
    c. language-design time

8. [2] Circle the correct operational semantics description of **for (**expr1**;** expr2**;** expr3**) {**statements**}**

```
        expr2
loop:   expr1
        if expr3 == 0
                goto out
        statements
        expr1
        goto loop
out:
```

```
        expr1
loop:   if expr2 == 0
                goto out
        statements
        expr3
        goto loop
out:
```

```
        expr1
loop:   statements
        expr3
        if expr2 == 0
                goto out
        goto loop
out:
```

9. [6] Which formalism for describing dynamic semantics would you use for each of these needs, and why?
    a. prove that some code is correct
    b. describe what machine code a compiler should generate for some specific control structure
    c. formally describe the semantics of some language construct

    Hint: Denotational semantics describes meaning in terms of compositional meanings of expressions, and before-after mappings specifying how the state changes after the execution of control structures; operational semantics describes the sequence of actions that would be performed by a machine implementing that construct; and axiomatic semantics involves making assertions, including conditional assertions about postconditions that depend on the values of preconditions.

10. [2] C++, unlike most languages, allows allocation of objects on the stack. Name one advantage and one disadvantage of this option.

/13

11. [3] Would you prefer static typing or dynamic typing,

    a. If you're designing a language to be highly readable?

    b. … writable?

    c. … efficient?

12. [2] Go has no classes, but it still has methods. Explain how this is possible.

13. [1] In the following C code, what is the type of pfun2?

```
int myfun2 (int x, int y) {return x + y};
int (*pfun2) (int in);
pfun2 = myfun2;
z = (*pfun2) (3, 4);
```

14. [2] Name one language in which parameters can be subprograms, and one language in which this is not possible.

15. [6] Fill in each blank with the appropriate letter

    a. not part of Java
    b. in Java, passed by reference
    c. in Java, passed by value
    d. in Java, passed by value-return
    e. in Java, only possible in non-recursive methods

    ___ function parameters
    ___ primitive types as parameters
    ___ objects and other reference types as parameters
    ___ array parameters

    Pick one of the parameter types above, and say how it is handled differently in some language other than Java.

16. [1] In Go, many functions return two arguments, with the second being non-nil if an error occurs, as in:

```
resp, err := http.Get(url)
if err != nil { // handle the error somehow
} else {   //process the response
```

Java has a different way of handling such things. What is it?

/15

17. [1] What is the Java keyword most similar to defer keyword in Go?

18. [1] What is the syntax for starting a coroutine (“goroutine”) in Go?

19. [2] Is a goroutine more like a thread or more like a process? Why?

20. [4] Briefly explain and illustrate any two of the following Rob Pike aphorisms.

   a. “language influences thought”
   b. “a language can have too many features”
   c. “readability is paramount”
   d. “simplicity is the art of hiding complexity”

21. [5] For each of the following fragments of Go code, indicate whether it will be

P a parse error (syntax error)
C another kind of compile-time error
R a runtime error
N never detected as an error by the system, but certainly not correct
V valid

   a. ___ **for i < 7 {update(i)}**
   b. ___ **<- ch**
   c. ___ **func addone (int i) int {return i++}**
   d. ___ **for {go addone(i)}**
   e. ___ **arrayl[:10]**

/13

22. [2] Under what conditions can an optimizing compiler safely change the order of evaluation of subexpressions in a Boolean expression?

23. [2] Under what conditions can an optimizing compiler safely change the order of evaluation of arguments in a function call?

24. [1] In Haskell, Prolog, Go and many other languages the single underscore (_) means the same thing. What is it?

25. [14 points] True or False

T F One way to get an overloaded subprogram name in Java is for a subclass to override a method of its superclass.
T F One way to get an overloaded subprogram name in Java is to define methods with the same name but different signatures.
T F The way that parameters are passed may depend on the data type.

T F In a pure object-oriented language there are no primitive types.
T F Some object-oriented languages provide access-control options for fields.
T F Some object-oriented languages provide access-control options for methods.
T F A class in Java that implements an interface will, by default, inherit code from that interface.
T F A subclass in Java inherits the method definitions of its superclass.
T F C++ lacks a garbage collector.

T F "Generic" is another word for a parameterized abstract data type.
T F In prolog **length([a,b], 3]** is a function call.
T F Pure functional programming eliminates side-effects.
T F Recursive subprograms can refer to static variables.
T F In a **while** loop, **continue** and **break** are synonyms (at least in Java and C++).

26. [3] If subroutineY is called several times in a program,

T F For each activation record (stack frame) the static link must point to the same place.
T F For each activation record (stack frame) the dynamic link must point to the same place.
T F For each activation record, the "return" address is a pointer to somewhere in the stack.

27. [1] What is the difference between **==** and **===** , for example in PHP and Javascript?

28. [2] (Matt) Given a reference to a nonlocal variable in a static-scoped language, how does the compiler find the correct definition?

29. [2] Given a reference to a nonlocal variable in a dynamic-scoped language, how does the compiler find the correct definition?

30. [6] Write Haskell code for the function substitute-splice, which takes three arguments, and substitutes the first for every occurrence of the second in the third. For example

```
substitute-splice([ 8,9], 3, [1,2,3,4])
-> [1,2,8,9,4]
```

31. [9] Name one language, other than C or Java, that you would choose if you needed to

    a. write code to reformat database query results for display on a webpage

    b. write code to plan sophisticated sets of financial transactions (selling options, buying futures, hedging, etc.) in a way that's provably valid with respect to the firm's strategy and policies

    c. write code to simulate the weather of the entire globe

    d. quickly write code find the largest prime less than 10000

    e. convince your cousin's niece (age 11) that programming can be easy

    f. write simple scripts (macros) to add new behaviors to Excel and other Office programs

    g. write apps (user-interface-intensive small applications) for iOS (Apple) devices

    h. write cloud (server) software

    i. write code for embedded microcontrollers, as in washing machines

/19