

CS-3160
Concepts of Programming Languages
Spring 2015

EXAM #1
(Chapters 1 - 6)

Name: _____

SCORES

MC: _____/75 PROB #1: _____/15 PROB #2: _____/10

TOTAL: _____/100

Multiple Choice Responses

Each multiple choice question in the separate packet is worth 3 points. There is no partial credit in this section. Choose the **BEST** answer of those given. Please indicate your answer by **DARKENING** the response that you have chosen **in the space provided below**. Make your selection obvious. Don't forget to mark your answers on this sheet. Only answers indicated below will be graded.

- | | | | | |
|--------------|---------------|---------------|---------------|---------------|
| 1. a b c d e | 6. a b c d e | 11. a b c d e | 16. a b c d e | 21. a b c d e |
| 2. a b c d e | 7. a b c d e | 12. a b c d e | 17. a b c d e | 22. a b c d e |
| 3. a b c d e | 8. a b c d e | 13. a b c d e | 18. a b c d e | 23. a b c d e |
| 4. a b c d e | 9. a b c d e | 14. a b c d e | 19. a b c d e | 24. a b c d e |
| 5. a b c d e | 10. a b c d e | 15. a b c d e | 20. a b c d e | 25. a b c d e |

Problem #1 (15 pts)

Consider the following grammar:

- $S \rightarrow (A \mid S \wedge A)$
- $A \rightarrow (B \mid B (* \mid /) A)$
- $B \rightarrow (C \mid C (+ \mid -) B)$
- $C \rightarrow \text{one of } \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

The operator semantics are the usual ones and are given in the table below. Circle the appropriate precedence (1 is highest) and associativity.

operator	name	example	Precedence	Associativity
+	addition	$6 + 2 = 8$	1 2 3 4 5	L R
-	subtraction	$6 - 2 = 4$	1 2 3 4 5	L R
*	multiplication	$6 * 2 = 12$	1 2 3 4 5	L R
/	division	$6 / 2 = 3$	1 2 3 4 5	L R
^	exponentiation	$6 ^ 2 = 36$	1 2 3 4 5	L R

Draw the parse tree, starting from **S**, for the following expression:

$$8 / 2 * 2 ^ 5 - 6 - 4 ^ 4 / 4 - 2$$

What is the value of this expression? _____

Problem #2 (10 pts)

A grammar consists of three binary operators and single-digit integer literals. The precedence and associativity are shown in the following table.

operator	Precedence	Associativity
#	1	R
@	2	L
\$	3	R

Develop a set of productions (BNF or EBNF) that reflects this grammar.

CS-3160 Multiple Choice Questions

Carefully record your response on the cover sheet of the exam.

- #1 The name used to describe having multiple ways to accomplish the same task is
- (a) redundancy.
 - (b) operator overloading.
 - (c) functional transparency.
 - (d) feature multiplicity.
 - (e) orthogonality.
- #2 When there are few restrictions on how primitive constructs can be combined, the language is said to be
- (a) orthogonal.
 - (b) flexible.
 - (c) tightly bound.
 - (d) simple.
 - (e) loosely constrained.
- #3 Which of the following are the three primary categories of languages, per the author?
- (a) declarative, functional, and logical.
 - (b) imperative, object-oriented, functional.
 - (c) imperative, declarative, and substantive.
 - (d) imperative, functional, and logical.
 - (e) function-oriented, data-oriented, object-oriented.
- #4 A general observation that is true more often than not regarding language design decisions is that
- (a) only decisions that improve all evaluation criteria should be adopted.
 - (b) improvement in one area often requires sacrifice in another.
 - (c) optimal decisions can be achieved only by carefully ranking evaluation criteria.
 - (d) decisions must reflect the practical realities faced by language implementers.
 - (e) no clear decisions can be made until all potential uses of the language are evaluated.
- #5 When two or more identifiers access the same memory cell, this is known as
- (a) aliasing.
 - (b) dereferencing.
 - (c) deferred access.
 - (d) cross referencing.
 - (e) inferred access.

- #6 The smallest parts of a language, such as operators, special words, numeric values, are called
- (a) lexemes.
 - (b) productions.
 - (c) sentences.
 - (d) tokens.
 - (e) phrases.
- #7 Categories of lexemes are called
- (a) productions.
 - (b) tokens.
 - (c) phrases.
 - (d) sentences.
 - (e) lexemes.
- #8 The 'syntax' of a programming language is
- (a) the collection of key words and the rules for how they may be used.
 - (b) the meaning of its expressions, statements, and program units.
 - (c) the rules for how the language's statements may be combined to create a program.
 - (d) the rules that govern the overloading of operators.
 - (e) the form of its expressions, statements, and program units.
- #9 The 'semantics' of a programming language is
- (a) the form of its expressions, statements, and program units.
 - (b) the collection of key words and the rules for how they may be used.
 - (c) the rules that govern the overloading of operators.
 - (d) the meaning of its expressions, statements, and program units.
 - (e) the rules for how the language's statements may be combined to create a program.
- #10 The language described by a grammar
- (a) is dictated by whether a left-most or a right-most derivation is specified.
 - (b) is 'context-free' if and only if it can be recognized by a finite automaton.
 - (c) is independent of the manner (order) in which rules are applied.
 - (d) is not context-free if it allows recursion.
 - (e) must be finite to have any practical utility (infinitely long programs are a theoretical curiosity).

- #11 A language recognizer performs what task?
- (a) Determines the order in which the productions can be applied to produce a particular sentence.
 - (b) Creates a hierarchical description of the grammatical structure of a sentence.
 - (c) Produces lower level instructions that implement the semantics of a sentence.
 - (d) Answers the question of whether a particular sentence is a member of a particular language.
 - (e) Reduces a sentence into a stream of tokens.
- #12 A parser performs what task?
- (a) Produces lower level instructions that implement the semantics of a sentence.
 - (b) Answers the question of whether a particular sentence is a member of a particular language.
 - (c) Creates a hierarchical description of the grammatical structure of a sentence.
 - (d) Determines the order in which the productions can be applied to produce a particular sentence.
 - (e) Reduces a sentence into a stream of tokens.
- #13 A lexer performs what task?
- (a) Produces lower level instructions that implement the semantics of a sentence.
 - (b) Answers the question of whether a particular sentence is a member of a particular language.
 - (c) Reduces a sentence into a stream of tokens.
 - (d) Creates a hierarchical description of the grammatical structure of a sentence.
 - (e) Determines the order in which the productions can be applied to produce a particular sentence.
- #14 A context free grammar imposes precedence of operators by placing
- (a) lower precedence non-terminals to the left of lower precedence non-terminals.
 - (b) lower precedence operators closer to the Start symbol in the production chain.
 - (c) lower precedence operators lower in the parse tree.
 - (d) higher precedence non-terminals to the left of lower precedence non-terminals.
 - (e) higher precedence operators closer to the Start symbol in the production chain.
- #15 The term "static" generally implies
- (a) that a variable is accessible even after the function that allocated it terminates.
 - (b) that a variable is global in scope.
 - (c) that all attributes, except the value, are known and fixed prior to runtime.
 - (d) that a variable is available to functions in other program units.
 - (e) that the associated attribute is known prior to runtime and does not change once the program has started running.

- #16 Which of the following is NOT true: Compared to dynamically scoped languages, statically scoped languages
- (a) execute faster.
 - (b) are more prevalent.
 - (c) are easier to read.
 - (d) are more reliable.
 - (e) require fewer parameters to be passed to functions.
- #17 The referencing environment of a statement is
- (a) the collection of all variables that are visible in that statement.
 - (b) the collection of all functions that can be referred to by the statement.
 - (c) the function in which the statement resides.
 - (d) the file in which the statement resides.
 - (e) the collection of all variable either read or written to within that statement.
- #18 In most languages, variables that are local to functions have what kind of storage binding?
- (a) Implicit heap-dynamic.
 - (b) Stack-dynamic.
 - (c) Elaborated.
 - (d) Static.
 - (e) Explicit heap-dynamic.
- #19 The scope of a variable is
- (a) the range of statements in which the variable is visible.
 - (b) fixed throughout the execution of the program.
 - (c) determined by the type of the variable.
 - (d) the range of values that the variable can assume.
 - (e) directly associated with the lifetime of the variable.
- #20 What distinguishes static (lexical) scoping from dynamic scoping?
- (a) In static scoping variables in the calling method are visible to the called method.
 - (b) Static scoping can be determined prior to execution while dynamic scoping can only be determined at run time.
 - (c) Dynamically scoped variables only exist for the lifetime of the program unit in which they are declared.
 - (d) Statically scoped variables exist for the entire execution of the program.
 - (e) In dynamic scoping all variables are visible to any method in the same class.

- #21 Primitive data types
- (a) are direct reflections of the underlying hardware representations.
 - (b) do not include character types since these may be defined in terms of ASCII integer values.
 - (c) are all based on Boolean types, which directly reflect hardware capabilities.
 - (d) are those data types that are not defined in terms of other data types.
 - (e) are limited to integer and floating point numeric types.
- #22 An enumeration type is generally
- (a) not supported by modern languages because of the lessened readability that usually results.
 - (b) a range of integers starting with zero.
 - (c) a set of named constants.
 - (d) not used except when working with explicit array limits.
 - (e) defined by the compiler based on the usage of integer literals.
- #23 A type whose variables may store different data types at different times is known as a
- (a) union.
 - (b) structure.
 - (c) class.
 - (d) array.
 - (e) record.
- #24 What does "lost memory" refer to?
- (a) A block of allocated memory that is no longer accessible to a user program.
 - (b) A pointer that refers to a heap-dynamic variable that has been deallocated.
 - (c) A pointer that has been set to nil (or NULL).
 - (d) Memory that can no longer be written to because it has been damaged.
 - (e) A pointer that points to memory containing data of a different type than the pointer was declared as.
- #25 Decimal data types are intended to
- (a) eliminate cumulative roundoff error in financial calculations.
 - (b) store string representations of decimal values.
 - (c) store decimal values, including fractional values, exactly over some range.
 - (d) enable fast hardware processing of financial data.
 - (e) store arbitrary precision decimal values.