1. Who is the author of the *primary* textbook for the class?  
   (a) Sibelius; (b) Sebesta; (c) Sethi; (d) Stansifer; (e) Scott
2. How many students received an ‘F’ from the instructor recently for reasons of academic misconduct?  
   (a) 2; (b) 20; (c) 200; (d) 2,000
3. true / false There will be a short quiz at the beginning of (almost) every class meeting.
4. In a class of 50 computer science seniors, one should expect how many to fail to put their name on the quiz card?  
   (a) 0; (b) 5; (c) 10; (d) 50
5. What does the common abbreviation IDE stand for (as used in Chapter 1 of the textbook)?
Quiz: Fri, 25 Aug 2017

1. Which of the several paths through the textbook by Scott will we be following?
   (a) plan “F”    (b) plan "R”    (c) plan "P”    (d) plan "C”

2. Abstraction means
   (a) signification of words or forms; (b) the medium of expression;
   (c) human-oriented presentation of data; (d) free from convoluted interactions;
   (e) act of determining essential properties.

3. Arabic is to linguistics as:
   (a) expression::visualization; (b) complexity::abstraction;
   (c) Python::programming languages; (d) gender::sex.

4. Software engineers need math because:
   (a) software consists of abstract constructs; (b) visualization of data is mathematical;
   (c) calculus is essential to calculating; (d) computer operations come from mathematics;

5. Which one of the following is not a computational paradigm:
   (a) logic programming; (b) imperative; (c) categorical; (d) functional.
1. The origin of the word “zero” comes from
   (a) Arabic; (b) Babylonian; (c) Catalan; (d) Dutch

2. The American name of the Greek letter τ rhymes with
   (a) adieu; (b) bureau; (c) Corfu; (d) [Charles] Dow

3. Which one of the following is not a scripting language:
   (a) Bash; (b) Perl; (c) Whitespace; (d) Python

4. In what language was the operating system Unix written?
   (a) Apl; (b) B; (c) C; (d) D
1. true / false  Declarative programming emphasizes the “how” over the “why.”
2. true / false  HTML is a programming language.
3. true / false  FORTRAN can reasonably be considered the first programming language.
4. true / false  Unlike a compiler, an interpreter stays around for the execution of the program.
5. true / false  ISO stands for the organization officially known as the International Standards Organization.
6. true / false  Machine languages are oriented toward the hardware and “high-level” languages are oriented toward humans.
1. true / false The syntax known as “Cambridge prefix” is used for all data and all code in LISP.

2. true / false Translation to native code can be done by an interactive system.

3. true / false Translation to native code can be done after execution begins.

4. true / false Java cannot be translated once and then executed over and over.

5. true / false An interactive language system is always an interpreter.

6. true / false Translation to byte-code is becoming more popular.
Quiz: Wed, 6 Sep 2017

1. Ada
2. APL
3. COBOL
4. C++
5. C
6. FORTRAN
7. Java
8. LISP
9. Pearl
10. Python

A. IBM, J. Backus
B. Augusta Ada Bryon
C. Dahl and Nygaard
D. Ralph Griswold
E. Kenneth Iverson
F. Guido van Rossum
G. John McCarthy
H. Kernighan and Richie
I. Bjarne Stroustrup
J. Sun, J. Gosling
K. US DoD, G. Hopper
L. US DoD, J. Icbaih
M. Larry Wall
Quiz: Fri, 15 Sep 2017

1. What is a formal language?
2. When is the Fortran project due?
3. What word best describes the view of syntax in each of the following courses?
   
   i. Formal Languages
   A. implementation
   B. representation
   C. expressivity
   D. recursion
   E. description

   ii. Progr Languages

   iii. Compiler Constr

4. true / false \(((a \cdot b) + (c \cdot d))^* \cdot c\) is a regular expression over the alphabet \(\Sigma = \{a, b, c, d, e\}\).
Quiz: Mon, 18 Sept 2017

What formal languages over the alphabet \{a, b, c, d\} do the following regular expression represent? Choose from the formal languages below. (You may choose a letter any number of times.)

1. \emptyset^*  
2. (a + b)^*  
3. (a^*)^*  
4. (a + \emptyset)^*  
5. (((a \cdot b) + (c \cdot d))  
6. (((a \cdot b) + (c \cdot d))^* \cdot c)  
7. ((a + b) + a^*) \cdot c)  
8. (a^* + b)^*  
9. ((a + b)^* + (a + c)^*)

A. \{\}  
B. \{\epsilon\}  
C. \{abcd\}  
D. \{ab, cd\}  
E. \{a, b, aa, ab, ba, bb, \ldots\}  
F. \{\epsilon, a, b, aa, ab, ba, bb, \ldots\}  
G. \{ac, bc, aac, abc, bac, bbc, \ldots\}  
H. \{c, abc, cdc, abcdc, 
       cdcdc, cdabc, ababc, \ldots\}  
I. none of the above
Quiz: Wed, 20 Sept 2017

1. true / false  It is possible to define what good syntax is for a programming language.

2. true / false  Syntax diagrams are equivalent to context-free grammars.

3. true / false  Back references can be defined in terms of the primitive regular expressions and, so, are just “macros” or “syntactic sugar.”

4. true / false  An ambiguous grammar and an unambiguous grammar may both describe the same language.

5. true / false  Regular expressions are great because they are more expressive than other common formalisms.

6. true / false  Scanner generators and parser generators are examples of a kind of programs which enable programmers to describe what they want and not how to implement it.

7. true / false  Lexical analysis determines the phrase structure of a language’s tokens.
Quiz: Fri, 22 Sep 2017

Test on Monday, 25 Sept 2017!
Syntax and Semantics

1. true / false  A formula of first-order logic can be used to characterize a set of computer states.

2. true / false  “Sue me if my postcondition is false, sue you if my precondition is false.”

3. true / false  The Cherokee script is used in writing FORTRAN programs.

4. true / false  Partial correctness means the program satisfies some of the postconditions.

5. Show that the following grammar with non-terminals $S$, $A$, and $l$ is ambiguous:

   $S \rightarrow A$
   $A \rightarrow A \times A \mid l$
   $l \rightarrow a \mid b \mid c$
Quiz: Fri, 22 Sep 2017

Test on Monday, 25 Sept 2017!
Syntax and Semantics

Fill in the box with the phrase that best describes the approach of each of the following types of semantics:

<table>
<thead>
<tr>
<th>Type of Semantics</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. denotational</td>
<td>A rules for evaluation</td>
</tr>
<tr>
<td>2. operational</td>
<td>B Post systems</td>
</tr>
<tr>
<td>3. natural</td>
<td>C rules relating states</td>
</tr>
<tr>
<td>4. structural</td>
<td>D attribute grammars</td>
</tr>
<tr>
<td>5. axiomatic</td>
<td>E mathematical objects</td>
</tr>
<tr>
<td></td>
<td>F “small-step” transitions</td>
</tr>
<tr>
<td></td>
<td>G an abstract machine</td>
</tr>
</tbody>
</table>