

Name: _____

1. (1 pt.)

- **Read all material carefully.**
- This test is **closed book, closed notes**.
- However, you may refer to **one** standard Letter-sized sheet of paper (both sides) that has **notes hand-written by you**. If used, these notes must **include your name** near the top and must be **submitted** along with this test.
- Computing or communication devices of any kind (laptop computers, tablets, phones, calculators, etc.) are not permitted.
- Network access of any kind (cell, voice, text, data, etc.) is not permitted.
- Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
- Use class and textbook conventions for notation, algorithmic options, etc.
- Write only in designated areas of this test. Do not separate the pages and do not include any additional paper.

Print your name clearly in the space provided above.
Do not write anything else on this page.

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

(Do not view any other pages.)

Do not write on this page.
(It is for use in grading only.)

Q	Full Score
1	1
2	4
3	5
4	5
5	5
6	5
7	5
8	5
9	5
10	10
total	45

2. (4 pts.) Consider the following regular expression which uses the textbook's syntax, except that it uses $\odot$ instead of $.$ (a dot) for better legibility:

$$(a + b*) \odot f + m$$

Provide a *fully parenthesized* version of this regular expression (using the same, textbook, syntax).

3. (5 pts.) Provide a regular expression in *Python syntax* that is equivalent to the regular expression of Question 2. The regular expression should use parentheses only where strictly necessary.

4. (5 pts.) Using Python syntax, provide a regular expression that matches a string iff (if and only if) it satisfies all the following requirements.

- It starts with the two-letter string **be**.
- It ends with the two-letter string **nd**.
- Between the above, it contains one or more instances of either the five-letter string **owwie** or the three-letter string **yay** (interleaved arbitrarily).

5. (5 pts.) In the context of *PLY*-based parsing, provide the definition of a function `p_3` corresponding to rule 3 of the grammar of Question 7. The function should set the value associated with the left-hand-side of the rule to the concatenation of the three-character string `R3:` and the values of the two items on the right-hand-side items. [Reminder: present code very clearly.]
6. (5 pts.) Using the facilities of the Python `re` module, provide a code snippet for determining whether the string value of a variable `s` matches the regular expression of Question 3, printing the string on a line by itself if it matches and a blank line otherwise.

7. (5 pts.) Consider the following context-free grammar (with parenthesized rule numbers on the right).

$$S \rightarrow \text{w o } R L \text{ d} \quad (1)$$

$$\rightarrow \text{d o } L R \text{ w} \quad (2)$$

$$R \rightarrow \text{a b} \quad (3)$$

$$| \text{ a } R \text{ b} \quad (4)$$

$$L \rightarrow \text{e} \quad (5)$$

$$| \text{ e e } L \quad (6)$$

Ignore all white space. **Clearly state** whether the following *sentence* is *valid* (belongs to the language of the grammar).

d o e e e a a b b w

If it is valid then provide a *leftmost derivation* for it **using classroom conventions** (in particular, underlining replaced symbols and annotating arrows with rule numbers); else explain (as precisely as possible) why it is not valid. [Hint: It may be easier to answer Question 8 first.]

8. (5 pts.) If the sentence of Question 7 is not valid then make as small a change as possible to the sentence to yield a valid sentence (else use the unchanged sentence here). In either case, provide a *parse tree* for the (original or modified) sentence.

9. (5 pts.) Repeat Question 7 using the following sentence instead of the one there:

w o a a b b e e e e e d

10. (5 pts.) Repeat Question 8 using the sentence of Question 9 instead of the one there.