Name:_____Key_____ Grade:_____ (of 10)

1. a. Which programming language(s) will be on the PAPER part of the exam?

   Python C (Both)

   b. Which programming language(s) will you use for the COMPUTER part of the exam?

   Python (C) Both

2. Where can you find a study guide for the final exam?

   **Course web site → Schedule page → Session 30**

   **→ Resources column**

   **→ Final Exam Topics and Sample Problems**

3. When you call a function, memory is allocated for each parameter and initialized to be a *copy* of the corresponding actual argument.

   (True) or False (circle your choice)

4. If the parameter is a pointer, changing the value of that *pointer* changes the value of the corresponding actual argument.

   True or (False) (circle your choice)

5. If the parameter is a pointer, changing the value of its *pointee* changes the value of the corresponding actual argument's *pointee*.

   (True) or False (circle your choice)

6. When you return from a function, the memory for its parameters and local variables is returned to the system, and hence those variables and the memory in which they were stored can no longer be referenced safely.

   (True) or False (circle your choice)

7. What standard library function do we use **in C** to dynamically allocate memory for data structures like arrays and structures?

   **malloc**

8. If we dynamically allocated memory to store an array containing 300 **`Dog`** instances (where **`Dog`** is a structure type), what would the ***declaration*** of the variable to reference that memory look like?

   ```
   Dog*  dogs;
   ```
   (any variable name is OK here and in the following problems)

9. Continuing the previous problem, what single statement would we use to give that variable an appropriate value?

   ```
   dogs = (Dog*) malloc(300 * sizeof(Dog));
   ```
   (make sure the two Dog phrases are just right – the 1st with an asterisk, the 2nd without)

10. Continuing the previous problem, assume that there is a ***`makeDog`*** function that returns a random instance of the **`Dog`** structure. Write a loop that initializes the 300 **`Dog`**'s referenced by your variable of the previous problem, to random **`Dog`**'s.

    ```
    for (k = 0; k < 300; ++k) {
        dogs[k] = makeDog();
    }
    ```

11. An object (in Python or any other object-oriented language):

    - Knows things about itself
    - Can be asked to do things
    - Can provide information about itself or other objects that it knows about

    Circle all of the above that apply.

12. A C structure instance:

    - Knows things about itself
    - Can be asked to do things
    - Can provide information about itself or other objects that it knows about

    Circle all of the above that apply.

13. Objects store information in _____**fields (aka instance variables)**_____.

14. Objects manipulate their information through ____**methods**_____.

15. Each object is an instance of some _____**class**_____.

16. Objects are created by calling _____**constructors**_____.

17. Suppose there is a Cat class whose constructor takes a string that is the name of the Cat. Write the **Python** statement that constructs a new Cat whose name is "Puss in Boots" and stores that Cat instance in the variable named ***shrekCat***.

    `shrekCat = Cat("Puss in Boots")`

18. In the definition of the Cat class, what is the name of the special function that the statement in the previous problem calls?

    `__init__`

19. Suppose that the Cat class has a field called **age**. Write the statement that sets ***shrekCat***'s age to 13.

    `shrekCat.age = 13`

20. In the definition of the Cat class, how do you recognize the fields?

    **They follow the expression `self.` (*self*, followed by a dot)**

21. Suppose that the Cat class has a method called **incrementAge**. Write the statement that increments ***shrekCat***'s age.

    `shrekCat.incrementAge()`

22. What value does the ***self*** parameter get when a constructor or method is called?

    **The value of the variable that preceded the dot in the constructor or method call. For example, if the call were**
    `shrekCat.incrementAge()`
    **then *self* in the *incrementAge* method would be set to the value of *shrekCat*.**

23. How long do the fields of an object persist?

    **Exactly as long as the object persists.**

24. Can a method of a class refer to all of the fields of the class? (Yes) or No
    (circle your choice)

25. Tell your instructor about anything from today's session (or from the course so far) that you found confusing or still have a question about. If none, please write "None".