

1. What are the two fundamental parts of an Object?

\_\_\_\_\_ & \_\_\_\_\_

Questions:

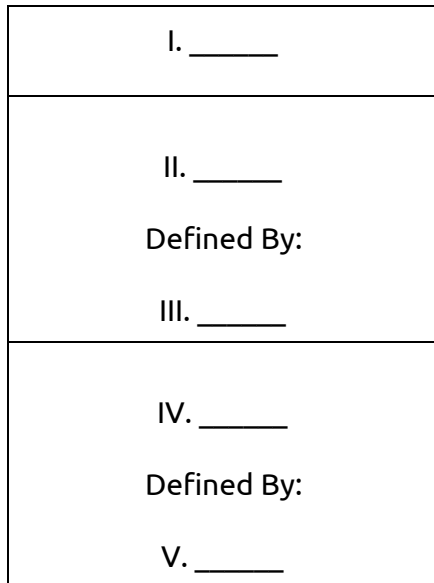
- 2. What is an Object? \_\_\_\_\_
- 3. What is a Class? \_\_\_\_\_
- 4. What is a method? \_\_\_\_\_
- 5. What is a field? \_\_\_\_\_
- 6. What is a constructor? \_\_\_\_\_  
\_\_\_\_\_

Definitions:

- A. A function with the same name as it's containing class that creates and sets up an Object
- B. A code blueprint for creating Objects
- C. Related variables and functions, encapsulated in a single data type
- D. A class variable
- E. A class function

7. Label the three sections of the UML Class Diagram for an Object:

Diagram:



Terms:

- A. Behavior
- B. Methods
- C. Fields
- D. Class Name
- E. State

8. Given the following Class definition, supply the appropriate answers:

```
public class Circle {
    int x, y, radius;
    PApplet p;
    Circle(int x, int y, int radius, PApplet p){
        this.p = p;
        this.x = x;
        this.y = y;
        this.radius = radius;
    }
    public void move(){
        if(y < 0){
            y = p.height;
        }
        y--;
    }
    public void render(){
        p.noFill();
        p.ellipse(x, y, radius, radius);
    }
}
```

A.

What are the names of the fields?

---

B.

What are the names of the methods?

---

C.

What are the types of the parameters that the constructor takes?

---

D.

Circle and Label the 'two parts of an Object' defined by this Class

9. Draw a UML Class Diagram for the Class defined in the previous question (#8)

10. What does a Class definition look like for the following scenario:

- a. A Class named 'Tree' containing
  - i. a field of type integer called 'numLeaves'
  - ii. a field representing an array of Leaf Classes called 'leafList'
  - iii. a field of type double called 'treeSize'
  - iv. a constructor that takes no arguments and initializes 'numLeaves' and 'treeSize' to zero
  - v. a method called 'grow' that returns nothing, takes no arguments, and increments both 'numLeaves' and 'treeSize' by one

11. Draw a UML Class Diagram for the Class defined in the previous question (#10), as well as:
- a. A Leaf Class with:
    - i. a field of type integer called 'leafColor'
    - ii. a constructor that takes no arguments
    - iii. a method called 'changeLeafColor' that returns an integer and takes no arguments
  - b. A Plant Class with:
    - i. a field of type boolean called 'needsSunlight'
    - ii. a constructor that takes no arguments
    - iii. a method called 'wilt' that returns a double and takes no arguments
  - c. The Tree Class should inherit from Plant Class and the Leaf Class should belong to the Tree Class

12. Write a function called 'oddSum' that:
  - a. returns an integer and takes no arguments
  - b. sums up all odd numbers from 0 to 50 using a loop
  - c. returns the value of the sum
  
13. Write a function called 'getMax'
  - a. returns an integer and takes an array of integers as an argument
  - b. loops through all the values in the array given and finds the maximum value
  - c. returns the maximum value
  
14. Write a function called 'computeAverage' that:
  - a. returns a double and takes an array of doubles as an argument
  - b. computes the average of all the values of the array given as an argument
  - c. returns the value of the average
  
15. Write a function called 'isEven' that:
  - a. returns an integer and takes a single integer argument
  - b. returns one if the argument is even, or zero if the argument is odd

16. Explain the purpose of using Objects and Classes:

17. Explain the purpose of writing functions:

18. Explain what the keyword 'static' means in the context of Java classes

19. Explain what a Java ArrayList is and what it can do

20. Explain what a Java HashMap is and what it can do

21. Explain what a Java Scanner is and what it can do