

# Memory Diagrams

When tracing code by hand, it's helpful to draw a picture to keep track of variables, methods, and objects. Memory diagrams represent the state of a program at a particular moment in time.

## Content Learning Objectives

*After completing this activity, students should be able to:*

- Describe primitive values and references in a memory diagram.
- Draw memory diagrams that have variables, arrays and objects.
- Summarize differences between variables, arrays, and objects.

## Process Skill Goals

*During the activity, students should make progress toward:*

- Leveraging prior knowledge and experience of other students. (Teamwork)



## Model 1 Team Roles

Decide who will be what role for today; we will rotate the roles each week. If you have only three people, one should have two roles. If you have five people, two may share the same role.

Manager:
Presenter:
Recorder:
Reflector:

### Questions (15 min)

**Start time:**

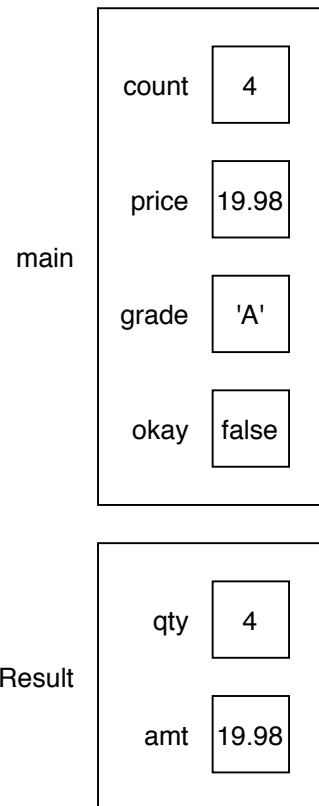
1. What is the difference between **bold** and *italics* on the role cards?
2. Manager: invite each person to explain their role to the team. Recorder: take notes of the discussion by writing down key phrases next to the table above.
3. What responsibilities do two or more roles have in common?
4. For each role, give an example of how someone observing your team would know that a person is not doing their job well.
  - Manager:
  - Presenter:
  - Recorder:
  - Reflector:

## Model 2 Local Variables

Consider the following example. The memory diagram shows the state of the program just before `printResults` returns for the second time:

```
public static void printResult(int qty, double amt) {
    System.out.printf("%d for $%.2f\n", qty, amt);
}

public static void main(String[] args) {
    int count = 3;
    double price = 9.99;
    char grade = 'A';
    boolean okay = true;
    printResult(count, price);
    count++;
    price *= 2;
    okay = !okay;
    printResult(count, price);
}
```



The output of the program is:

```
3 for $9.99
4 for $19.98
```

### Questions (15 min)

**Start time:**

5. How many variables are declared ...

a) in main?

b) in printResult?

6. How many times is each variable assigned?

a) count

d) okay

b) price

e) qty

c) grade

f) amt

7. Is there a small box for each declaration or each assignment? Justify your answer.

8. What do the six small boxes in the memory diagram represent?

9. What do the two large boxes in the memory diagram represent?

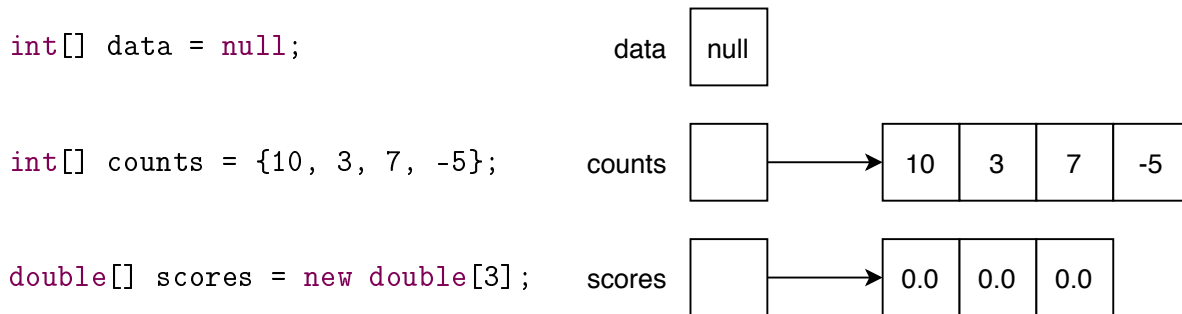
10. Why does the diagram indicate that `count` is 4 and `price` is 19.98, even though the source code says that `count = 3` and `price = 9.99`?

11. Based on the source code:

- a) Which method is defined first?
- b) Which method is executed first?

## Model 3 Arrays

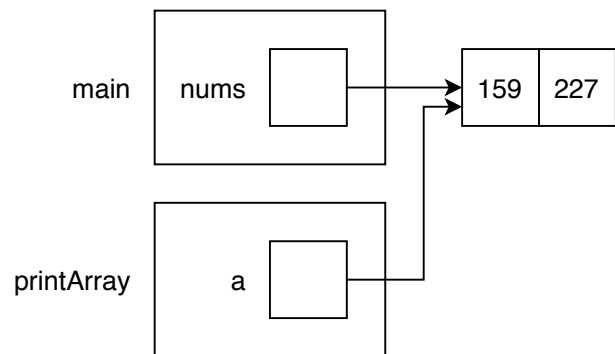
An array variable stores a *reference* to an array object. We draw references as arrows, because they “point” to other memory locations.



When passing an array to a method, only the reference is copied:

```
public static void printArray(int[] a) {  
    System.out.print("{ " + a[0]);  
    for (int i = 1; i < a.length; i++) {  
        System.out.print(", " + a[i]);  
    }  
    System.out.println("}");  
}
```

```
public static void main(String[] args) {  
    int[] nums = {159, 227};  
    printArray(nums);  
}
```



### Questions (15 min)

Start time:

12. What is the length of each array?

a) counts?

c) nums?

b) scores?

d) a?

13. Looking at both diagrams above:

a) How many array variables were declared?

b) How many array objects were created?

14. Based on counts and scores, describe two ways that array objects can be created. How are these two ways different from each other?

15. If the `printArray` method were to modify the array contents, would that change be visible in the main method? Explain your reasoning.

16. Draw a diagram of the following source code:

```
int[] data = {1, 2, 3};  
int[] copy = data;
```

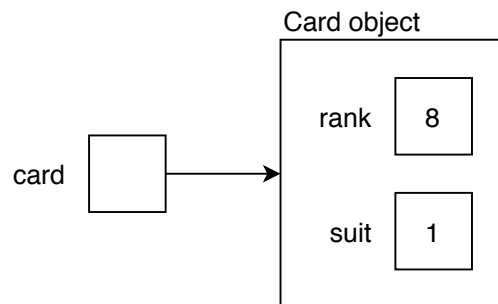
## Model 4 Objects

Consider the definition for a playing card:

```
public class Card {  
    private int rank; // 1=Ace, ..., 11=Jack, 12=Queen, 13=King  
    private int suit; // 0=Clubs, 1=Diamonds, 2=Hearts, 3=Spades  
  
    public Card(int rank, int suit) {  
        this.rank = rank;  
        this.suit = suit;  
    }  
}
```

Here is a memory diagram of a Card object:

```
Card card = new Card(8, 1);
```



### Questions (15 min)

**Start time:**

- Which card (i.e., “the \_\_\_\_\_ of \_\_\_\_\_”) is represented in the diagram?
- In one line of code, show how you would construct the “4 of Clubs”.
- What is the difference between lowercase `card` and uppercase `Card`? Explain in a few sentences how these concepts are illustrated in the diagram.
- How are arrays and objects similar? How are arrays and objects different? Explain your answer in terms of how they are drawn in memory diagrams.

**21.** Draw a diagram of the following source code:

```
Card card = null;
```

**22.** Draw a diagram of the following source code:

```
Card card = new Card(5, 2);
```

```
Card copy = card;
```