# CS 149

Professor: Kevin Molloy
(adapted from slides originally developed by Alvin Chao)

# Review of Java Primitive Types

| Keyword | Size (bytes) | Min Value | Max Value |
|---|---|---|---|
| Byte | 1 | -128 | 127 |
| Short | 2 | -32,768 | 32,767 |
| int | 4 | $-2^{31}$ | $2^{31} - 1$ |
| long | 8 | $-2^{63}$ | $2^{63} - 1$ |
| float | 4 | $\pm 3.4 \times 10^{-38}$ | $\pm 3.4 \times 10^{38}$ |
| double | 8 | $\pm 1.7 \times 10^{-308}$ | $\pm 1.7 \times 10^{308}$ |
| boolean | (depends) | false | true |
| char | 2 | '\u0000' | '\uffff' |

# Reference Types in JAVA

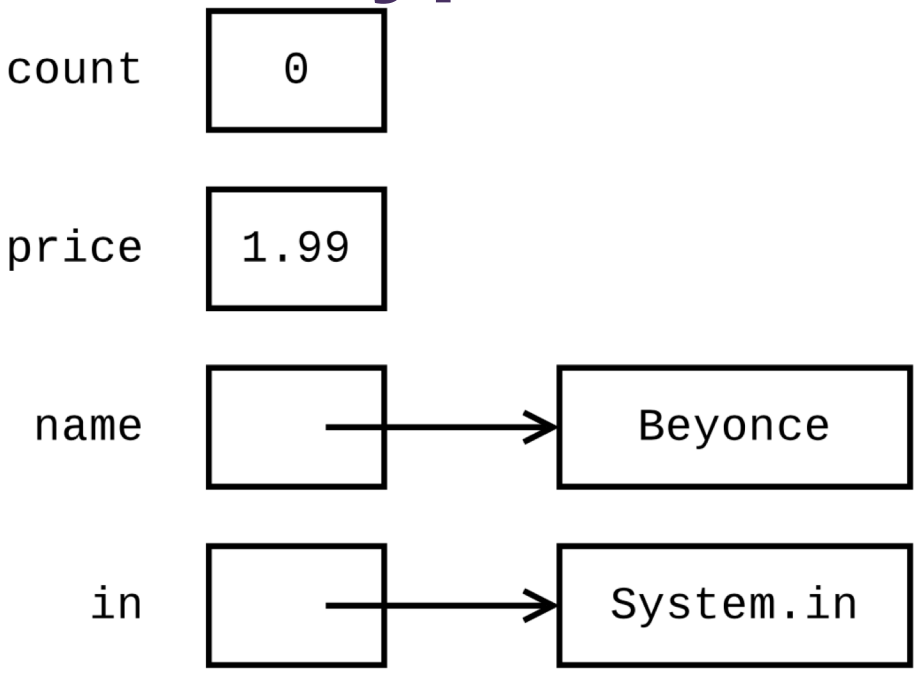
- A **reference** is a variable that refers so something else and can be used as an alias for that something else.

- In general, anything that is not one of the **primitive** types is a reference.

# Reference Types 1

```
int count;
double price;
String name;
Scanner in;

count = 0;
price = 1.99;
name = "Beyonce";
in = new Scanner(System.in);
```

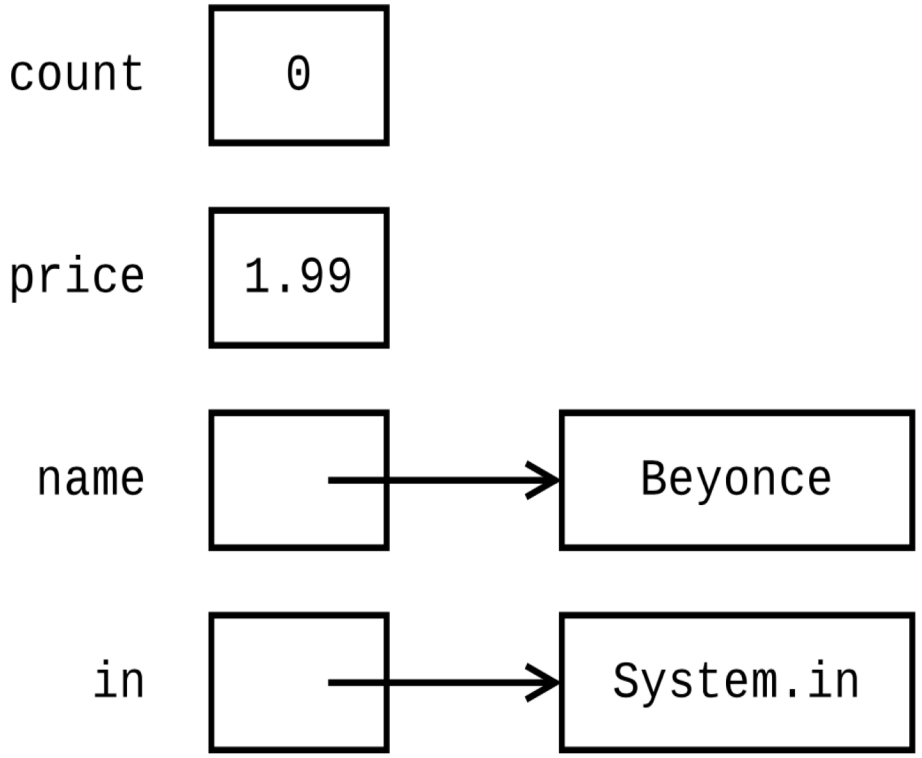



1. What are the reference types in the example above?
2. What is the difference between primitive and reference type names?
3. Variables in Java can use at most eight bytes of memory. The values "Beyonce" and System.in cannot be stored directly in the memory locations for name and in.
4. What is the value of the variable count? What is the value of the variable price?

# Reference Types 2

```
int count;
double price;
String name;
Scanner in;

count = 0;
price = 1.99;
name = "Beyonce";
in = new Scanner(System.in);
```




1. What is the value of the variable **name**?
2. What is the value of the variable **in**?
3. Carefully explain what it means to assign one variable to another. For example, what does the statement price = count; do in terms of memory?

# Ref Type Model

Draw a memory diagram for the following code.

```
int width;
double score;
Scanner input;
String first;
String other;
width = 20;
score = 0.94;
input = new Scanner(System.in);
first = "Taylor";
score = width;
other = first;
```

What is the output of the following statements after running the code above? Explain your answer using the diagram.

What happens if we run the following (express in our memory diagram and the output):

```
first = "Swift";

System.out.println(other);
```

- **Acknowledgements**
- Parts of this activity are based on materials developed by Helen Hu and Urik Halliday, modified by Chris Mayfield and Nathan Sprague, and licensed under CC BY-NC 4.0 International

---

</end>