Data Types

Java supports two main types of data: ***primitive types*** like `int` and `double` that represent a single value, and ***reference types*** like `String` and `Scanner` that represent more complex data.

Manager: Recorder:

Presenter: Reflector:

Content Learning Objectives

After completing this activity, students should be able to:

- Explain how using roles improves the team's success.
- Name Java's primitive data types and give examples of each one.
- Identify illegal assignment statements, and explain why they are illegal.
- Describe what it means for variables to store a reference to an object.

Process Skill Goals

During the activity, students should make progress toward:

- Providing feedback on how well other team members are working. (Teamwork)

Copyright © 2021 Chris Mayfield and Stoney Jackson. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Meta Activity: Team Disruptions

Common disruptions to learning in teams include: talking about topics that are off-task, teammates answering questions on their own, entire teams working alone, limited or no communication between teammates, arguing or being disrespectful, rushing to complete the activity, not being an active teammate, not coming to a consensus about an answer, writing incomplete answers or explanations, ignoring ideas from one or more teammates.

Questions (10 min) Start time:

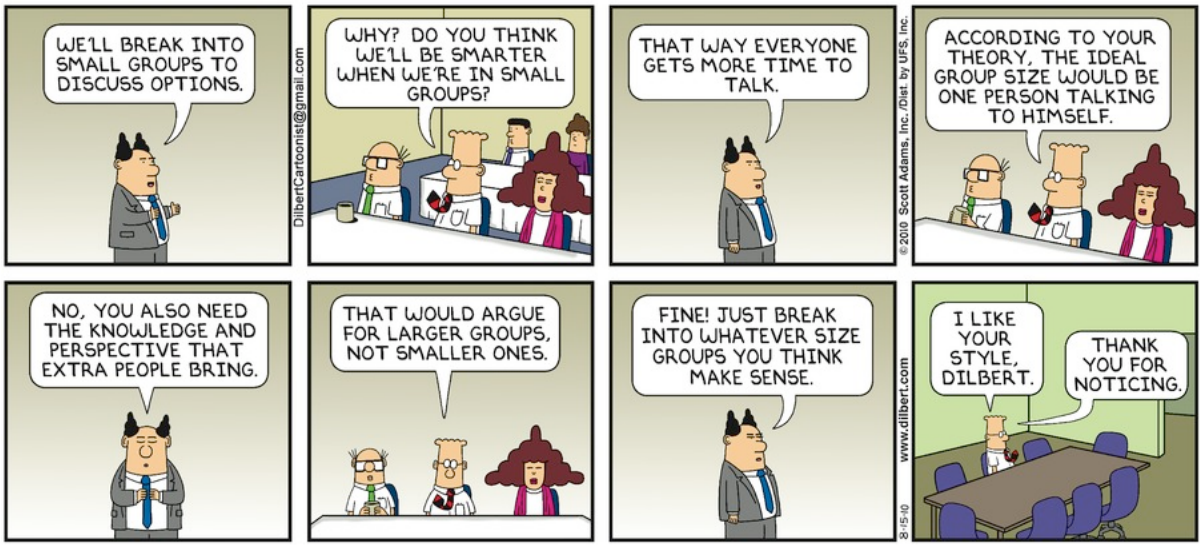
1. Pick four of the disruptions listed above. For each one, find something from the role cards that could help improve the team's success. Use a different role for each disruption.

a) Manager:

b) Presenter:

c) Recorder:

d) Reflector:



Dilbert by Scott Adams. © Andrews McMeel Syndication. http://dilbert.com/strip/2010-08-15

Model 1 Primitive Types

Keyword	Size	Min Value	Max Value	Example
`byte`	1 byte	-128	127	`(byte) 123`
`short`	2 bytes	$-32,768$	$32,767$	`(short) 12345`
`int`	4 bytes	-2^{31}	$2^{31}-1$	`1234567890`
`long`	8 bytes	-2^{63}	$2^{63}-1$	`123456789012345L`
`float`	4 bytes	-3.4×10^{38}	3.4×10^{38}	`3.14159F`
`double`	8 bytes	-1.8×10^{308}	1.8×10^{308}	`3.141592653589793`
`boolean`	1 byte	N/A	N/A	`true`
`char`	2 bytes	0	65,535	`'A'`

Note that 1 byte is 8 bits, i.e., eight "ones and zeros" in computer memory. Since there are only two possible values for each bit, you can represent $2^8 = 256$ possible values with 1 byte.

Questions (15 min) Start time:

2. Which of the primitive types are integers? Which are floating-point?

3. Why do primitive types have ranges of values? What determines the range of the data type?

4. Why can't computers represent every possible number in mathematics? Will they ever be able to do so?

5. Since a `byte` can represent 256 different numbers, why is its max value 127 and not 128?

6. What is the data type for each of the following values?

1.14159	7.2E-4	-128
0	0.0	'0'
-1.0F	-13L	false
123	'H'	true

7. Based on the examples below, when does Java allow you to assign one type of primitive variable to another?

```
int int_ = 3;
long long_ = 3L;
float float_ = 3.0F;
double double_ = 3.0;

int_ = int_;
int_ = long_;      // illegal
int_ = float_;     // illegal
int_ = double_;    // illegal

long_ = int_;
long_ = long_;
long_ = float_;    // illegal
long_ = double_;   // illegal
```

```
float_ = int_;
float_ = long_;
float_ = float_;
float_ = double_;  // illegal

double_ = int_;
double_ = long_;
double_ = float_;
double_ = double_;

int_ = '0';
int_ = false;      // illegal
double_ = '0';
double_ = false;   // illegal
```

8. Given the following variable declarations, which of the assignments are not allowed?

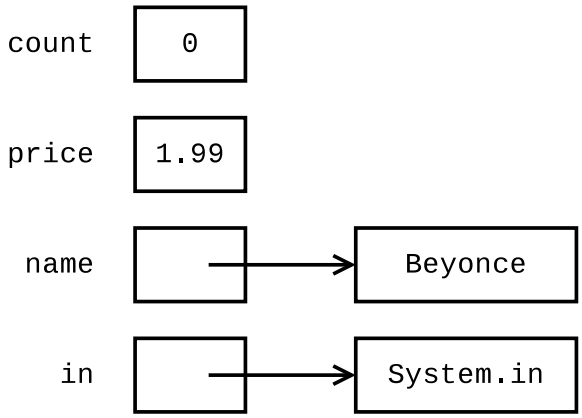
```
byte miles;
short minutes;
int checking;
long days;
float total;
double sum;
boolean flag;
char letter;
```

```
checking = 56000;
total = 0;
sum = total;
total = sum;
checking = miles;
sum = checking;
flag = minutes;
days = '0';
```

Model 2 Reference Types

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

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



Java has eight primitive types (see Model 1). All other types of data are called ***reference*** types, because **their value is a memory address**. When drawing memory diagrams, use an arrow to reference other memory locations (rather than make up integer values for the actual addresses).

Questions (20 min) Start time:

9. What are the names of the reference types in the example above?

10. Notice the pattern Java uses for type names like `int` and `String`:

a) Are reference type names all lowercase or capitalized?

b) Are primitive type names all lowercase or capitalized?

11. Variables in Java can use at most eight bytes of memory. Explain why the values `"Beyonce"` and `System.in` cannot be stored directly in the memory locations for `name` and `in`.

12. What is the value of the variable `count`? What is the value of the variable `price`?

13. What is the value of the variable `name`? What is the value of the variable `in`?

14. Carefully explain what it means to assign one variable to another. For example, what does the statement `price = count;` do in terms of memory?

15. Draw a memory diagram for the following code. Make sure your answer is consistent with what you wrote for #14.

```
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;
```

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

```
first = "Swift";
System.out.println(other);
```

17. (Optional) Paste the contents of *TaylorSwift.java* into Java Visualizer. What differences do you notice between the diagram in Java Visualizer and yours from #15?