

Primitives and References

Primitive Data Types

- Primitive data types

type	desc.	Default value (for fields)
byte	8-bit signed two's complement integer	0
short	16-bit signed two's complement integer	0
int	32-bit signed two's complement integer	0
long	64-bit signed two's complement integer; 256L	0L
float	single-precision 32-bit IEEE 754 floating point; 3.14f	0.0f
double	double-precision 64-bit IEEE 754 floating point; 3.14d	0.0d
boolean	true or false	false
char	16-bit Unicode character (UTF-16); '\u0000' ~ '\uffff'	'\u0000'

- Strings: `String greeting = "Hello world!";`
 - Default value: `null`

Integer Literals

- Decimal: base 10
 - `int decVal = 26;`
- Hexadecimal: base 16
 - `int hexVal = 0x1a;`
- Binary: base 2
 - `int binVal = 0b11010;`

Floating-Point Literals

- Single-precision
 - `float f1 = 123.4f;`
 - `float f2 = 1.234e2f;`
 - `float f3 = (float)123.4;`
- Double-precision
 - `double d1 = 123.4d;`
 - `double d2 = 1.234e2d;`
 - `double d3 = 123.4;`
 - `double d4 = 1.234e2;`

Character & String Literals

- Character

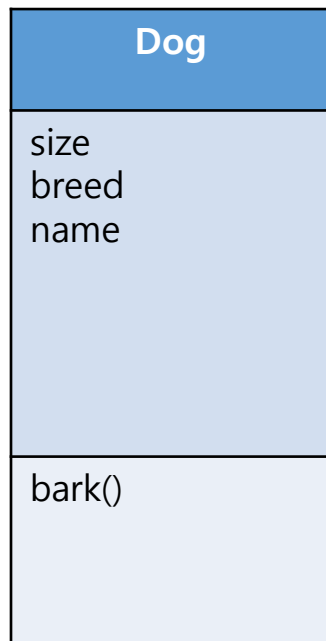
- `char C_with_circumflex = 'Ŵu0108'; // Ĉ`
- `char c = 'c';`

- String

escape sequence	meaning	escape sequence	meaning
<code>\b</code>	(backspace)	<code>\"</code>	" (double quote)
<code>\t</code>	(tab)	<code>\'</code>	' (quote)
<code>\n</code>	(newline)	<code>\\</code>	Ŵ (backslash)
<code>\f</code>	(form feed)		
<code>\r</code>	(carriage return)		

Class & Object

[class]



[class name]

[instance variables]
knows
state

[methods]
does
behavior

Dog : class

a dog : object (instance of Dog)

The *size* of a dog is big.

A dog *barks*.

References

```
Dog terry = new Dog();
```

Declare a
variable

Link the
object to the
reference
variable

Create an
object.

References

- There is **no** object variable in **Java**.
 - *Dog c_terr;* *// in C++*
- In Java, there are only object **reference** variables
 - *Dog java_terr = new Dog();* *// in Java*
 - *Dog *c_pterr = new Dog();* *// in C++*
- *references* in Java != *references* in C++
 - References in Java are equivalent to pointers in C++.

References

```
Dog terry = new Dog("Terry");  
Dog baal = new Dog("Baal");  
Dog theDog = terry;  
theDog = baal;
```

```
if (theDog == terry) {  
    // The dog is Terry.  
} else if (theDog == baal) {  
    // The dog is Baal.  
}
```

References

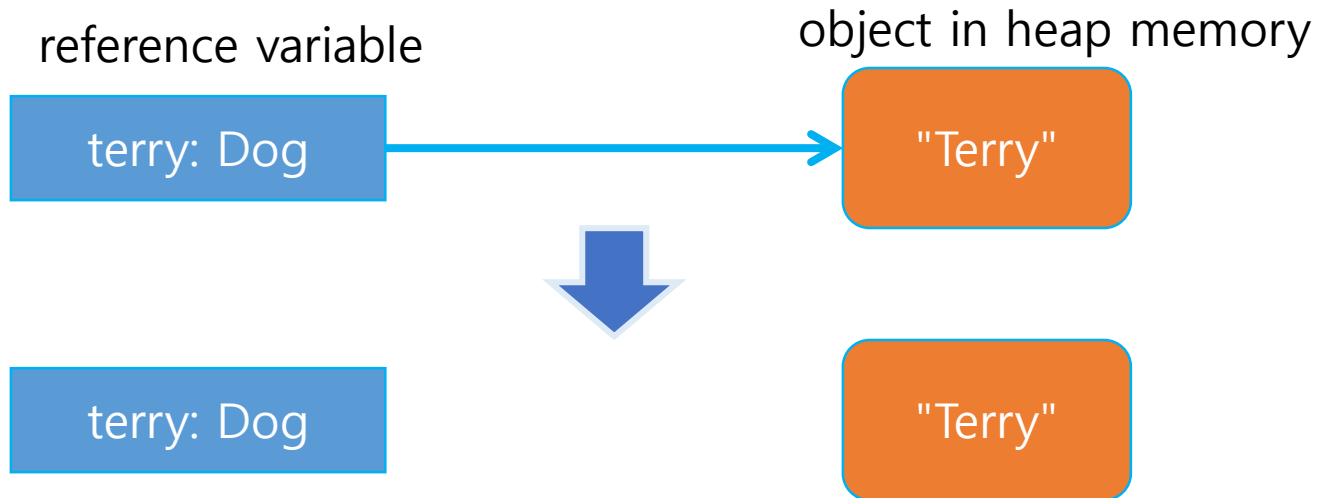
```
Dog terry = new Dog("Terry");  
Dog missingDog = new Dog("Terry");
```

```
if (terry == missingDog) {  
    // Terry is the dog I'm looking for.  
} else {  
    // They are different.  
}
```

References

- *null* reference value.

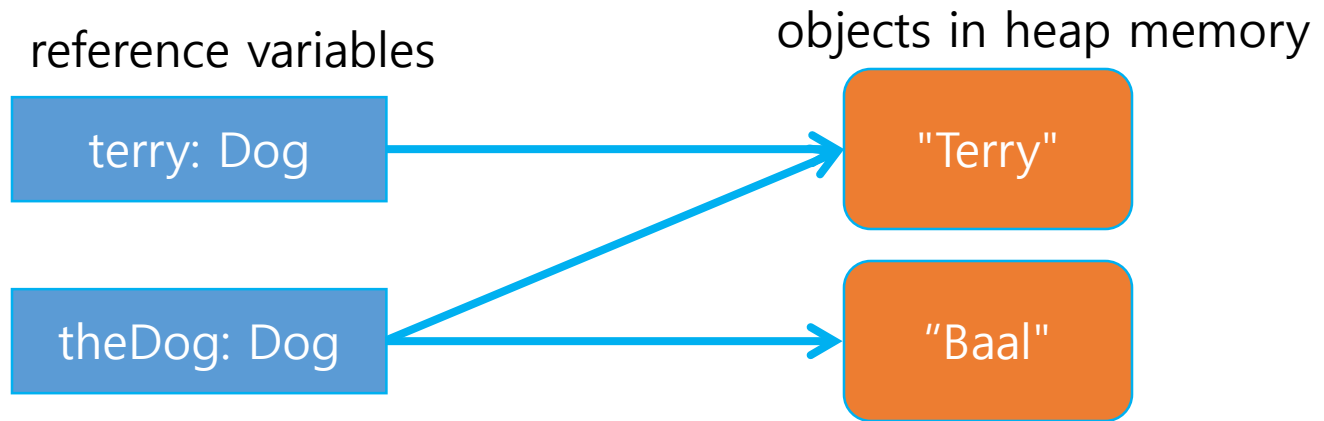
```
Dog terry = new Dog("Terry");  
terry = null;
```



Life on the heap in Java

- Garbage Collector (GC)
 - deletes unreferenced objects.

```
Dog terry = new Dog("Terry");  
Dog theDog = new Dog("Baal");  
theDog = terry;
```



Array (1)

```
int[] heights = new int[3];  
heights[0] = 3; heights[1] = 2; heights[2] = 4;  
  
for (int i = 0; i < heights.length; i++) {  
    System.out.println(heights[i]);  
}  
  
String[] names = {"Harry", new String("Mark"), "Happy"};  
  
for (String name : names) {  
    System.out.println(name);  
}  
  
int[] anotherHeights = heights; // Arrays are objects.
```

Array (2)

```
Dog[] pets = new Dog[7];
```

```
pets[0] = new Dog();  
pets[1] = new Dog();  
pets[2] = new Dog();  
pets[3] = new Dog();  
pets[4] = new Dog();  
pets[5] = new Dog();  
pets[6] = new Dog();
```

```
Dog[] pets = { new Dog(), new Dog(), new Dog(), new Dog(),  
new Dog(), new Dog(), new Dog() };
```

References

- Kathy Sierra and Bert Bates, *Head First Java*, O'Reilly, 2005.
- The Java Tutorials: Primitive Data Types
 - <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- The Java 7 SE API Specification: String
 - <http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

Q&A

class Object

- In java, all classes implicitly extend *Object*.

Object
<pre>#clone(): Object #finalize(): +equals(obj: Object): boolean +getClass(): Class<?> +hashCode(): int +notify(): +notifyAll(): +toString(): String +wait(): +wait(timeout: long): +wait(timeout: long, nanos: int):</pre>