

Week 5, Lecture 2

Exploring Methods



1

Announcements

- The deadline for the first three lab assessments is **TODAY**.
- Assignment 1 has been marked. Your marks and commented code can be accessed via the course web page.
- See the web page for **Challenge Problems**.

2

This Week

Java Genesis:

–Ch6: Methods

Lab Assessment 4 (deadline Week 8)

Quick Quiz for Chapter 5

3

Primitive types (e.g. integer)



The value of the variable **x** is the integer stored in the location associated with **x**.

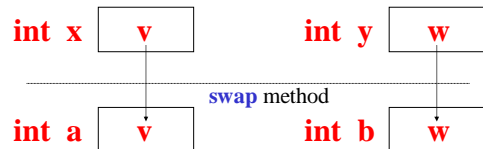
4

trying to swap primitive values

```
....  
int x = 5;  
int y = 21;  
swap(x, y);  
....  
public static void swap (int a, int b) {  
    int temp = a;  
    a = b;  
    b = temp;  
}
```

5

passing arguments of primitive type



When **swap** is invoked the integer values of **x** and **y** are copied to new locations. The values in these new locations are then swapped.



6

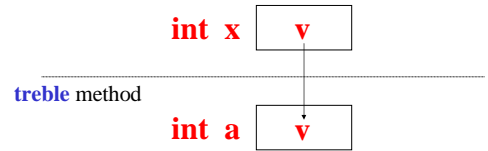
trebling primitive values

```
int x = 5;
treble(x);
....
public static void treble (int a) {
    a = 3*a;
}

--improved code-----
int x = 5;
x = treble(x);
....
public static int treble (int a) {
    return 3*a;
}
```

7

passing arguments of primitive type (cont)



When `treble` is invoked the integer value of `x` is copied to a new location. The value in this new location is then trebled.

`int a` `3v`

8

Object type (e.g. array)

`int [] xs` → `V0, V1, ..`

The value of the variable `xs` is the storage location where the elements (integers) in the array are stored. We can think of `xs` as naming the array.

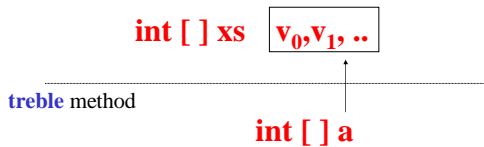
9

trebling array values

```
....
int [] xs = {5, 11, 3};
treble(xs);
....
public static void treble (int [] a) {
    for (int i=0; i<a.length; i++) {
        a[i] = 3*a[i];
    }
}
```

10

passing arguments of object type



When `treble` is invoked location `xs` is now directly referenced (named) by `a`. Hence `a` is an alias for `xs`. Each element in the referenced array is now trebled, so effectively each element in the array `xs` is trebled.

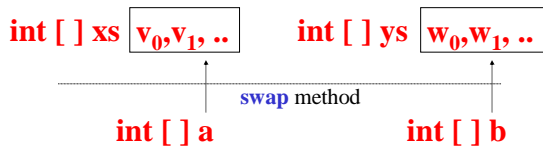
11

Trying to swap array values

```
....
int [] xs = {5, 11, 13};
int [] ys = {4, 10, 12};
swap(xs, ys);
....
public static void swap (int [] a, int [] b) {
    int [] temp = a;
    a = b;
    b = temp;
}
```

12

passing arguments of object type (cont)



When `swap` is invoked locations `xs` and `ys` are now directly referenced (named) by `a` and `b`. Hence `a` is an alias for `xs`; `b` is an alias for `ys`. If these references are swapped `a` references `ys` and `b` references `xs` but the contents of the arrays are not swapped: they must be swapped element by element.

13

swapping array values

```
....  
int [ ] xs = {5, 11, 13};  
int [ ] ys = {4, 10, 12};  
swap(xs, ys);  
....  
public static void swap (int [ ] a, int [ ] b) {  
    int temp;  
    for (int i=0; i<a.length; i++) {  
        temp = a[i];  
        a[i] = b[i];  
        b[i] = temp;  
    }  
}
```

14

Summing an Array

If `a` is an array of numbers, then sum of elements in `a`

$$= a[0] + \text{sum of elements in tail}(a)$$

15

for example....

If `a` is the array `{3, 7, 5}`, the sum of the elements in `a`

$$= a[0] + \text{sum of elements in tail}(a)$$

$$= 3 + \text{sum of elements in } \{7, 5\}$$

16