

CSE 21

Intro to Computing II

Lecture 3 – Methods (2)

Today

▶ Methods (2)

▶ Lab

- Lab 3 assigned this week (2/4 – 2/10)
 - Generic cheese shop
- Due in one week
 - Make sure to show your work to YOUR OWN TA (or me) before submission
 - Not **required** but **highly encouraged** to make sure you receive full credit

▶ Reading Assignment

- Sections 6.1 – 6.11 (including participation activities)
 - Work on the Participation Activities in each section to receive participation grade at the **end of semester** (based on at least 80% completion)

Extra Credit

- ▶ Up to 5 percentage points of total grade
- ▶ Based on completion of **challenge activities** of reading assignment sections
 - 20% complete = 1% of total grade
 - 40% complete = 2% of total grade
 - 60% complete = 3% of total grade
 - 80% complete = 4% of total grade
 - 100% complete = 5% of total grade
- ▶ Like participation activity, scores evaluated at **the end of semester**

3 processes of MOS Preference (review)

A. Get sample size input from user

B. Get samples

- For each person
 - Ask for choice (gather information)
 - Use tally counters

C. Output

- Ask if they would like to see a detailed count
 - If yes, display the tally counters
 - If no, then nothing is displayed

PreferenceMOS.java (review)

```
import java.util.Scanner;
```

```
public class PreferenceMOS {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Enter the total number of students: ");
```

```
        int max = input.nextInt();
```

```
        int android, ios, other, choice;
```

```
        android = ios = other = choice = 0;
```

```
        System.out.println("Preference? Android (1), iOS (2), or Both (3).");
```

```
        for (int i = 0; i < max; i++) {
```

```
            System.out.print("Enter choice: ");
```

```
            choice = input.nextInt();
```

```
            if (choice == 1) android++;
```

```
            else if (choice == 2) ios++;
```

```
            else if (choice == 3) {
```

```
                android++;
```

```
                ios++;
```

```
            } else other++;
```

```
        }
```

```
        System.out.print("See detailed count? Yes (1), or No (0): ");
```

```
        int detailed = input.nextInt();
```

```
        if (detailed == 1) {
```

```
            System.out.println("Prefer Android = " + android);
```

```
            System.out.println("Prefer iOS = " + ios);
```

```
            System.out.println("Prefer Other = " + other);
```

```
        }
```

```
    }
```

```
}
```

A

B

C

PreferenceMOSv2.java (review)

```
import java.util.Scanner;
```

```
public class PreferenceMOS {
```

```
    public static int MAX; // Sample size is a global variable shared by all methods in this class
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        int tally[] = new int[3]; // Use array instead of individual variables for simplicity  
        tally[0] = tally[1] = tally[2] = 0; // tally[0]: android, tally[1]: iOS, tally[2]: other
```

```
        getSampleSize(input); A
```

```
        System.out.println("Preference? Android (1), iOS (2), Both (3), or Other (4).");
```

```
        tallyCounter(input, tally, MAX); B
```

```
        seeCount(input, tally); C
```

```
    }
```

```
}
```

Methods



Why Methods? (review)

- ▶ Readability

- Succinct
- Organization

- ▶ Benefits

- Independent testing of sub-tasks
- Reusable code
 - Design and test a method once, and re-use it whenever you need to solve a similar problem
- Isolation from unintended side effects
 - The only variables from the caller that can be seen from a method are those in the argument list

- ▶ Think about a factory with different assembly lines.

Methods (review)

public static void main(String[] args)

Accessible
by Everyone

One per
Class

Returns
Nothing

Name

Array of
Arguments

public static int[] tallyCounter(**Scanner** in, int[] tally, int max)

Returns an
integer
array

Three Arguments: first of type
Scanner, second of type integer
array, third of type integer

Compile Error (review)

```
public class SimpleExample {  
    // Method Declaration like variables (callee)  
    public static void intro() {  
        System.out.println("Hi, my name is Santosh");  
    }  
    public static void main(String[] args) {  
        intro(2); // Method invocation (caller)  
    }  
}
```

#3

#4

#1

#2

#5



Flow of program

Giving an integer argument but callee is expecting no arguments

of arguments and Types have to match

Sum Example

```
public class PreferenceMOSv2{
```

tally[0] = 13

tally[1] = 18

```
// Method Declaration like variables (callee)
```

```
public static int CombinedTally(int num1, int num2) {
```

#3

```
    System.out.println("First tally is " + num1);
```

#4

```
    System.out.println("Second tally is " + num2);
```

#5

```
    int total = num1 + num2;
```

#6

```
    return total;
```

#7

```
}
```

```
public static void main(String[] args) {
```

#1

```
...
```

```
int sum #8 = CombinedTally(tally[0], tally[1]); // (caller)
```

#2

```
System.out.println("Total tally is " + sum);
```

#9

```
}
```

```
}
```

tally[0]: #(prefer android)

tally[1]: #(prefer iOS)

Sum Example

```
public class PreferenceMOSv2{
```

tally[0] = 13

tally[1] = 18

```
// Method Declaration like variables (callee)
```

```
public static int CombinedTally(int num1, int num2) {
```

#3

```
    System.out.println("First tally is " + num1);
```

#4

```
    System.out.println("Second tally is " + num2);
```

#5

```
    int total = num1 + num2;
```

#6

```
    return total;
```

#7

```
}
```

Local variables
for **total** only

```
public static void main(String[] args) {
```

#1

```
...
```

```
int sum #8 = CombinedTally(tally[0], tally[1]); // (caller)
```

#2

```
System.out.println("Total tally is " + sum);
```

#9

```
}
```

```
}
```

Output:

First tally is 13

Second tally is 18

Total tally is 31

Sum Usage

- ▶ Want to add 3 numbers (tally[0], tally[1], tally[2])
- ▶ First Option
 - `int total1 = CombinedTally(tally[1], tally[2]);`
 - `int total = CombinedTally(tally[0], total1);`
- ▶ Second Option (Substitution)
 - `int total = CombinedTally(tally[0], CombinedTally(tally[1], tally[2]));`
- ▶ Third Option (Commutative +)
 - `int total = CombinedTally(CombinedTally(tally[1], tally[2]), tally[0]);`

Sum Example: Scope

```
public class PreferenceMOSv2{  
    // Method Declaration like variables (callee)  
    public static int CombinedTally(int num1, int num2) {  
        System.out.println("First tally is " + num1);  
        System.out.println("Second tally is " + num2);  
        int total = num1 + num2;  
        return total;  
    }  
  
    public static void main(String[] args) {  
        ...  
        int num1 = 18, num2 = 13;  
        System.out.println("Main num1 is " + num1);  
        System.out.println("Main num2 is " + num2);  
        int total #11 = CombinedTally(num2, num1);  
        System.out.println("Sum is " + total);  
    }  
}
```

#6
#7
#8
#9
#10

#1

#2
#3
#4
#5
#12

// (caller switched
// arguments)

Sum Example: Scope

```
public class PreferenceMOSv2{
```

```
// Method Declaration like variables (callee)
```

```
public static int CombinedTally(int num1, int num2) { #6
    System.out.println("First tally is " + num1); #7
    System.out.println("Second tally is " + num2); #8
    int total = num1 + num2; #9
    return total; #10
}
```

Output: Main num1 is 18
Main num2 is 13
First tally is 13
Second tally is is 18
Sum is 31

```
public static void main(String[] args) { #1
    ...
    int num1 = 18, num2 = 13; #2
    System.out.println("Main num1 is " + num1); #3
    System.out.println("Main num2 is " + num2); #4
    int total #11 = CombinedTally(num2, num1); // caller switched #5
                                           // arguments
    System.out.println("Sum is " + total); #12
}
```

```
}
```

Sum Example: Scope

```
public class PreferenceMOSv2{  
    // Method Declaration like variables (callee)  
    public static int CombinedTally(int num1, int num2) {  
        System.out.println("First tally is " + num1);  
        System.out.println("Second tally is " + num2);  
        int total = num1 + num2;  
        num1 = 100; ←————— No Effect: Logical Error  
        return total;  
    }  
}
```

Two sets of variables:
num1, **num2** and **total** local to each
method are completely independent!

```
public static void main(String[] args) {  
    ...  
    int num1 = 18, num2 = 13;  
    int total = CombinedTally(num2, num1); // (caller)  
    System.out.println("Main num1 is " + num1);  
    System.out.println("Main num2 is " + num2);  
    System.out.println("Sum is " + total);  
}  
}
```

Multiple Returns (1)

tally[0] = 5

tally[1] = 10

```
public static int maxAndroidIOS(int num1, int num2) {  
    if (num1 > num2)  
        return num1;  
    if (num2 > num1)  
        return num2;  
    if (num2 == num1)  
        return num2;  
    return 0;  
}
```

#3

#4

#5

#6

```
public static void main(String[] args) {
```

#1

...

```
    int maxNumber #7 = maxAndroidIOS(tally[0], tally[1]);
```

#2

```
    System.out.println("Max is " + maxNumber);
```

#8

```
}
```


Multiple Returns (2)

tally[0] = 15

tally[1] = 10

```
public static int maxAndroidIOS(int num1, int num2) {           #3
    if (num1 > num2)                                             #4
        return num1;                                           #5
    if (num2 > num1)
        return num2;
    if (num2 == num1)
        return num2;
    return 0;
}

public static void main(String[] args) {                         #1
    ...
    int maxNumber #6 = maxAndroidIOS(tally[0], tally[1]);      #2
    System.out.println("Max is " + maxNumber);                 #7
}
```

Multiple Returns (3)

tally[0] = 20

tally[1] = 20

```
public static int maxAndroidIOS(int num1, int num2) {  
    if (num1 > num2)  
        return num1;  
    if (num2 > num1)  
        return num2;  
    if (num2 == num1)  
        return num2;  
    return 0;  
}  
  
public static void main(String[] args) {  
    ...  
    int maxNumber #8 = maxAndroidIOS(tally[0], tally[1]);  
    System.out.println("Max is " + maxNumber);  
}
```

#3

#4

#5

#6

#7

#1

#2

#9

Multiple Returns Optimized (if)

tally[0] = 5

tally[1] = 10

```
public static int maxAndroidIOSv1(int num1, int num2) {           #3
    if (num1 > num2)                                              #4
        return num1;
    return num2;                                                #5
}

public static void main(String[] args) {                          #1
    ...
    int maxNumber #6 = maxAndroidIOSv1(tally[0], tally[1]);      #2
    System.out.println("Max is " + maxNumber);                  #7
}
```

Multiple Returns (Conditional)

tally[0] = 5

tally[1] = 10

```
public static int maxAndroidIOSv2(int num1, int num2) {           #3
    return num1 > num2 ? num1:num2;                               #4
}                                                                True False
```

```
public static void main(String[] args) {                           #1
    ...
    int maxNumber #5 = maxAndroidIOSv2(tally[0], tally[1]);       #2
    System.out.println("Max is " + maxNumber);                   #6
}
```