CSE 21
Intro to Computing II

Lecture 2 – Review of CSE 20 (2)
Methods (1)

Today

- Review of CSE20 (2) and Methods (1)
- Lab
  - Lab 2 assigned this week (1/28 – 2/3)
    - Cheese shop
  - Due in one week
    - Make sure to show your work to YOUR OWN TA (or me) before submission
- 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)

Problem Statement (review)

- We want to survey the type of mobile OS students prefer. It will ask for a sample size and inquire for each person whether they like Android, iOS or both. It should report a breakdown of the data upon request.

Steps (review)

- Get input from user
  - How? Use a Scanner
  - What? To begin with, the sample size
- Get samples
  - For each person (for loop)
    - Ask for choice (gather information): 1 for Android, 2 for iOS, 3 for both (any other input tallied under "other").
    - Use tally counters:
      If choice is 1, android++
      2, ios++
      3, android++ and ios++
- Output

Final Code (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);
        }
    }
}
```

Problem Statement

- We want to survey the type of mobile OS students prefer. It will ask for a sample size and inquire for each person whether they like Android, iOS or both. It should report a breakdown of the data upon request.
- What if we want to keep track of all students' choices
  - Use an array!

Get number of students

```
import java.util.Scanner;
public class PreferenceMOSRecord {
        public static void main(String[] args) {
                Scanner input = new Scanner(System.in);
                System.out.print("Enter the total number of students: ");
                final int MAX = input.nextInt();
                int choices[] = new int[MAX];
        }
}
```

choices

[0]	[1]	[2]	[3]	[4]	…	…	…	…	[MAX - 1]

Looping

Original code

```
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++;
}
```

New code

```
for (int i = 0; i < MAX; i++) {
        System.out.print("Enter choice: ");
        choices[i] = input.nextInt();
        if (choices[i] == 1)
                android++;
        else if (choices[i] == 2)
                ios++;
        else if (choices[i] == 3) {
                android++;
                ios++;
        } else
                other++;
}
```

What if?

- Want to count students who exclusively prefer Android but not iOS
  - Remember the variable android is incremented for choices of both (1) and (3)

Count exclusive preference for Android

```
int i;
int androidOnly = 0;
for (i = 0; i < choices.length; i++) {
        if (choices[i] == 1)
                androidOnly++;
}
```

What if?

- Want to see if there are any students who prefer both Android and iOS

Combo

```
boolean preferBoth= false;
for (i = 0; i < choices.length; i++) {
        if (choices[i] == 3) {
                preferBoth = true;
                break; // Just need one person
        }
}
if (!preferBoth)
        System.out.println("No one prefers both!");
```

PreferenceMOS.java

```
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);
        }
    }
}
```

MOS Preference (3 processes)

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

```
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();                                          A
        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++;                                    B
            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);              C
            System.out.println("Prefer iOS = " + ios);
            System.out.println("Prefer Other = " + other);
        }
    }
}
```

PreferenceMOSv2.java

```
import java.util.Scanner;
public class PreferenceMOS {
        public static int MAX;
        public static void main(String[] args) {
                Scanner input = new Scanner(System.in);

                int tally[] = new int[3]; // Using int array instead of individual tally counters
                tally[0] = tally[1] = tally[2] = 0;

                getSampleSize(input);

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

                tallyCounter(input, tally, MAX);

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

Methods

A

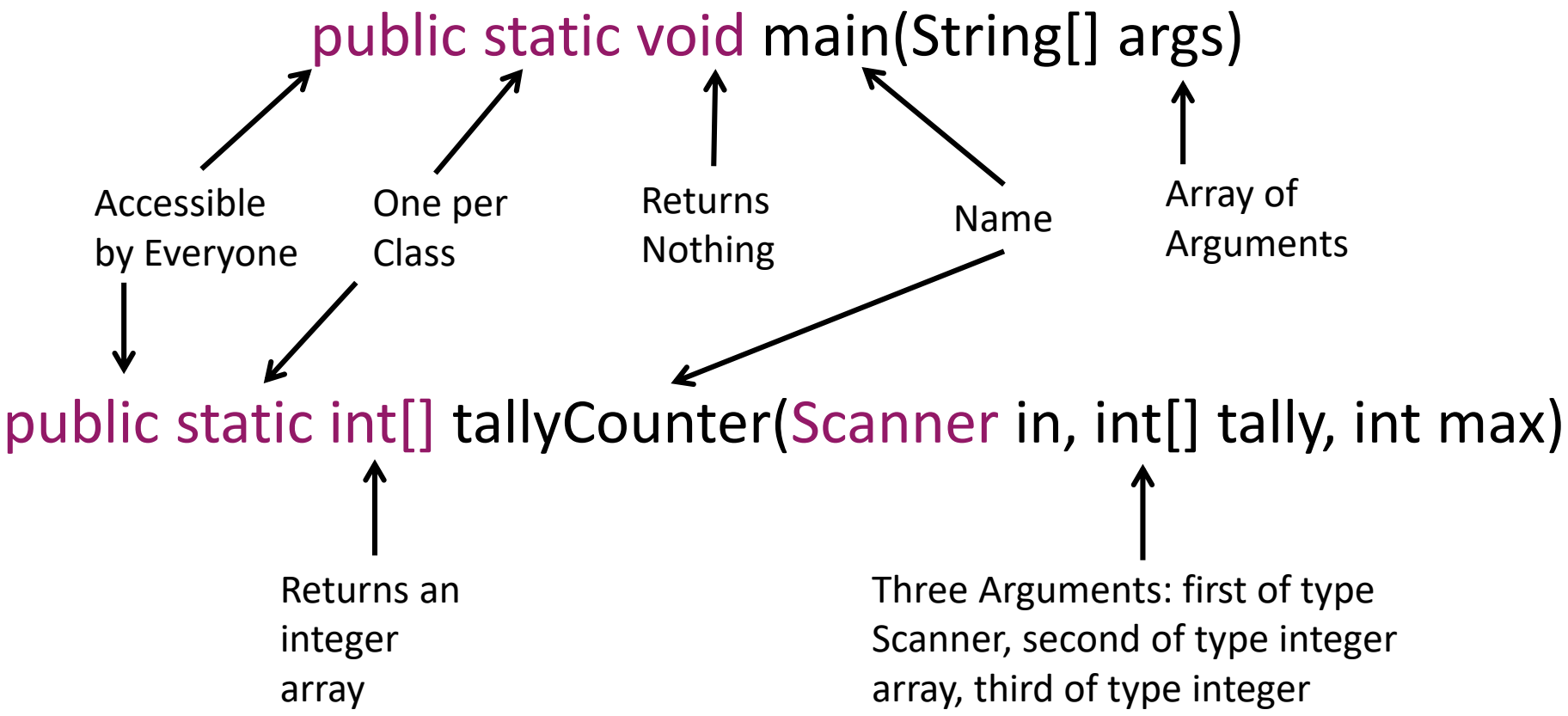
B

C

Why Methods?

- 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


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

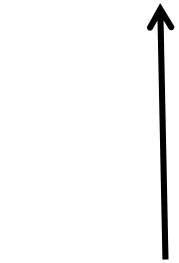
Simple Example

```
public class SimpleExample {
    // Method Declaration like variables (callee)
    public static void intro() {                          #3
        System.out.println("Hi, my name is Santosh");     #4
    }

    public static void main(String[] args) {              #1
        intro(); // Method invocation (caller)            #2
                                                          #5
    }
}
```

OUTPUT:

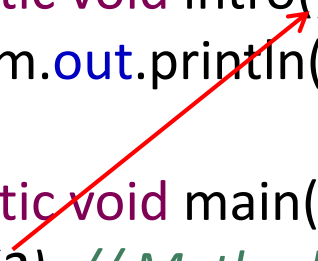
Hi, my name is Santosh



Flow of program

Compile Error

```
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)
    }
}
```



Giving an integer argument but callee is expecting no arguments

# of arguments and Types have to match