CSE 21
Intro to Computing II

Lecture 2 – Methods

Announcement

- Lab 1 due before start of next lab
- Reading assignment
  - Chapter 6.1 to 6.11 of textbook

CheeseShop.java

```java
public static void main(String[] args) {
Scanner input = new Scanner(System.in);

System.out.println("We sell 3 kinds of
   Cheese");
System.out.println("Dalaran Sharp: $1.25 per
   pound");
System.out.println("Stormwind Brie: $10.00 per
   pound");
System.out.println("Alterac Swiss: $40.00 per
   pound");
System.out.print("Enter the amount of Sharp:
   ");

int sharp = input.nextInt();
System.out.print("Enter the amount of Brie:
   ");
int brie = input.nextInt();
System.out.print("Enter the amount of Swiss:
   ");
int swiss = input.nextInt();

System.out.print("Display the itemized list?
   (1 for yes) ");
int itemized = input.nextInt();

double total = sharp*1.25 + brie*10.0 +
swiss*40.00;
```

```java
double discount = 0;
if (total >= 100)
  discount = 25;
else if (total >= 50)
  discount = 10;

if (itemized == 1) {
  if (sharp > 0)
  System.out.println(sharp + " lbs of Sharp
     @ $1.25 = $" + sharp*1.25);
  if (brie > 0)
  System.out.println(brie + " lbs of Brie @
     $10.00 = $" + brie*10.0);
  if (swiss > 0)
  System.out.println(swiss + " lbs of Swiss
     @ $40.00 = $" + swiss*40.00);
}

System.out.println();
System.out.println("Sub Total: $" + total);
System.out.println("-Discount: $" +
   discount);
System.out.println("Total    : $" + (total-
   discount));
}
```

Cheese Shop

A. List all the cheese types available and the prices

B. Asks the user how many pounds of each type of cheese to purchase

C. Calculate Sub Total (price*amount of each cheese added together)

D. Discount of Sub Total
   - for a $10 discount if their purchase is $50 or over
   - an additional $15 discount ($25 total) if $100 or over

E. Ask the user if they would like to see a list of what they purchased
   - If yes, a list comes up showing how much of each type of cheese they bought and the cost of each cheese
   - Display only the cheese they actually bought
   - If no then no itemized information is displayed

F. Display Sub Total, Discount and Total Price

CheeseShop.java

```java
public static void main(String[] args) {
Scanner input = new Scanner(System.in);
```

A

```java
System.out.println("We sell 3 kinds of
   Cheese");
System.out.println("Dalaran Sharp: $1.25 per
   pound");
System.out.println("Stormwind Brie: $10.00 per
   pound");
System.out.println("Alterac Swiss: $40.00 per
   pound");
System.out.print("Enter the amount of Sharp:
   ");
```

B

```java
int sharp = input.nextInt();
System.out.print("Enter the amount of Brie:
   ");
int brie = input.nextInt();
System.out.print("Enter the amount of Swiss:
   ");
int swiss = input.nextInt();
```

C

```java
System.out.print("Display the itemized list?
   (1 for yes) ");
int itemized = input.nextInt();

double total = sharp*1.25 + brie*10.0 +
swiss*40.00;
```

D

```java
double discount = 0;
if (total >= 100)
  discount = 25;
else if (total >= 50)
  discount = 10;
```

E

```java
if (itemized == 1) {
  if (sharp > 0)
  System.out.println(sharp + " lbs of Sharp
     @ $1.25 = $" + sharp*1.25);
  if (brie > 0)
  System.out.println(brie + " lbs of Brie @
     $10.00 = $" + brie*10.0);
  if (swiss > 0)
  System.out.println(swiss + " lbs of Swiss
     @ $40.00 = $" + swiss*40.00);
}
```

F

```java
System.out.println();
System.out.println("Sub Total: $" + total);
System.out.println("-Discount: $" +
   discount);
System.out.println("Total     : $" + (total-
   discount));
}
```

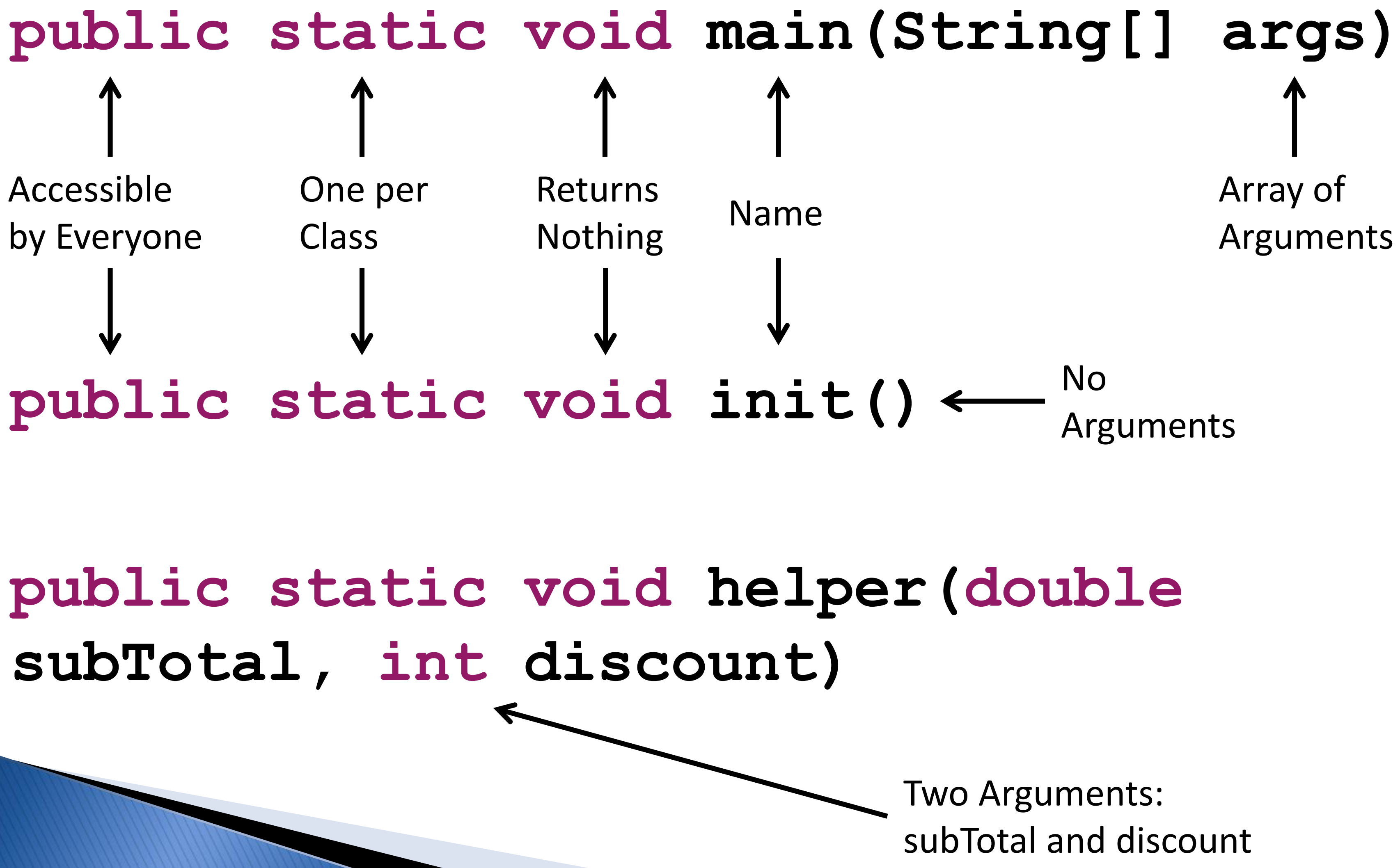
CheeseShopV2.java

```
public static void main(String[] args) {
  Scanner input = new Scanner(System.in);

  intro();  A

  int sharp = getAmount(input, "Sharp");
  int brie = getAmount(input, "Brie");        B
  int swiss = getAmount(input, "Swiss");

  double total = calcSubTotal(sharp, brie, swiss);     C

  System.out.print("Display the itemized list? (1 for yes) ");
  int itemized = input.nextInt();
  if (itemized == 1)
    itemizedList(sharp, brie, swiss);    E

  System.out.println();
  printTotal(total,discount(total));
}     F           D
```

Methods Motivation

- 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

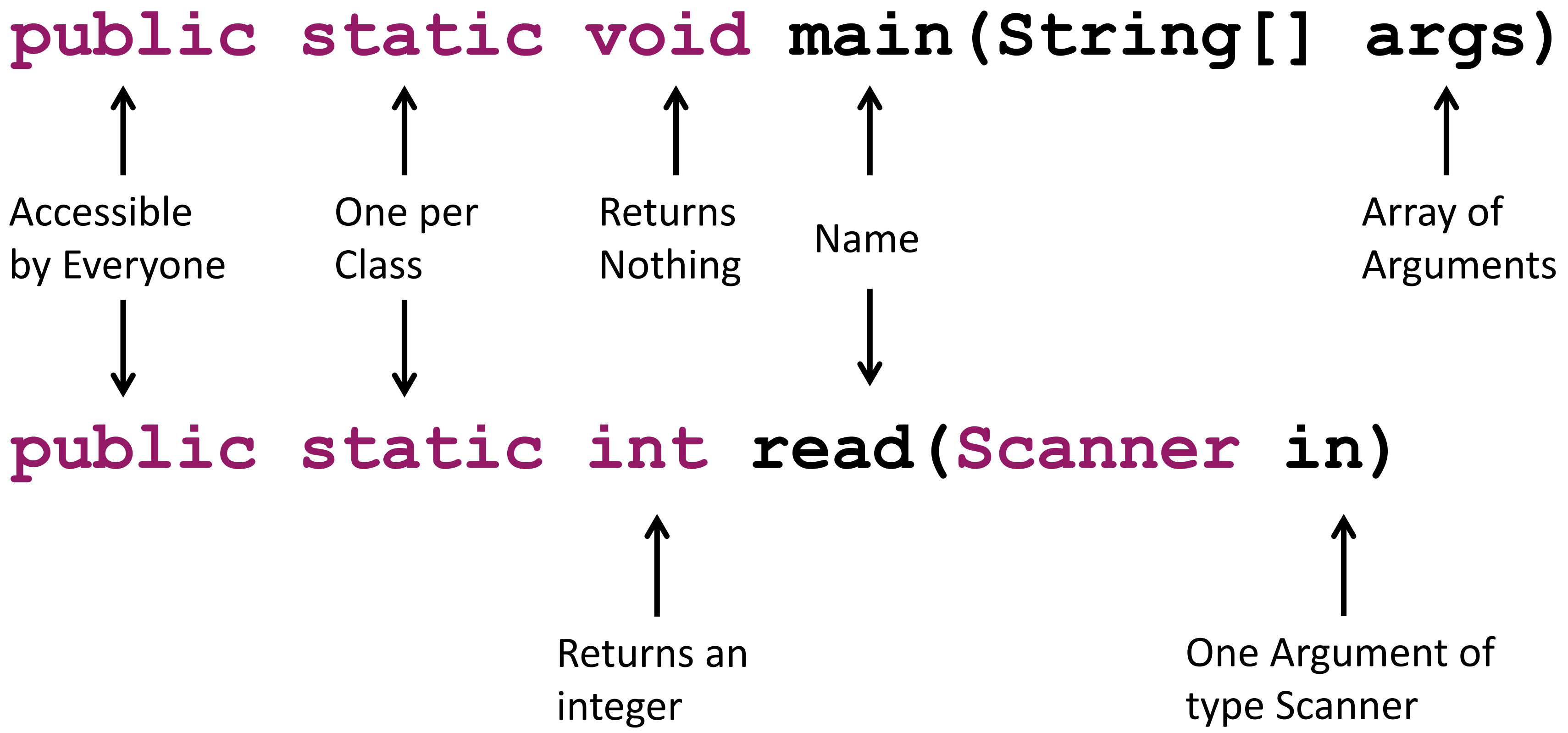
```
public static void init()
```

No
Arguments

```
public static void helper(double
subTotal, int discount)
```

Two Arguments:
subTotal and discount

Methods



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

Accessible by Everyone

One per Class

Returns Nothing

Name

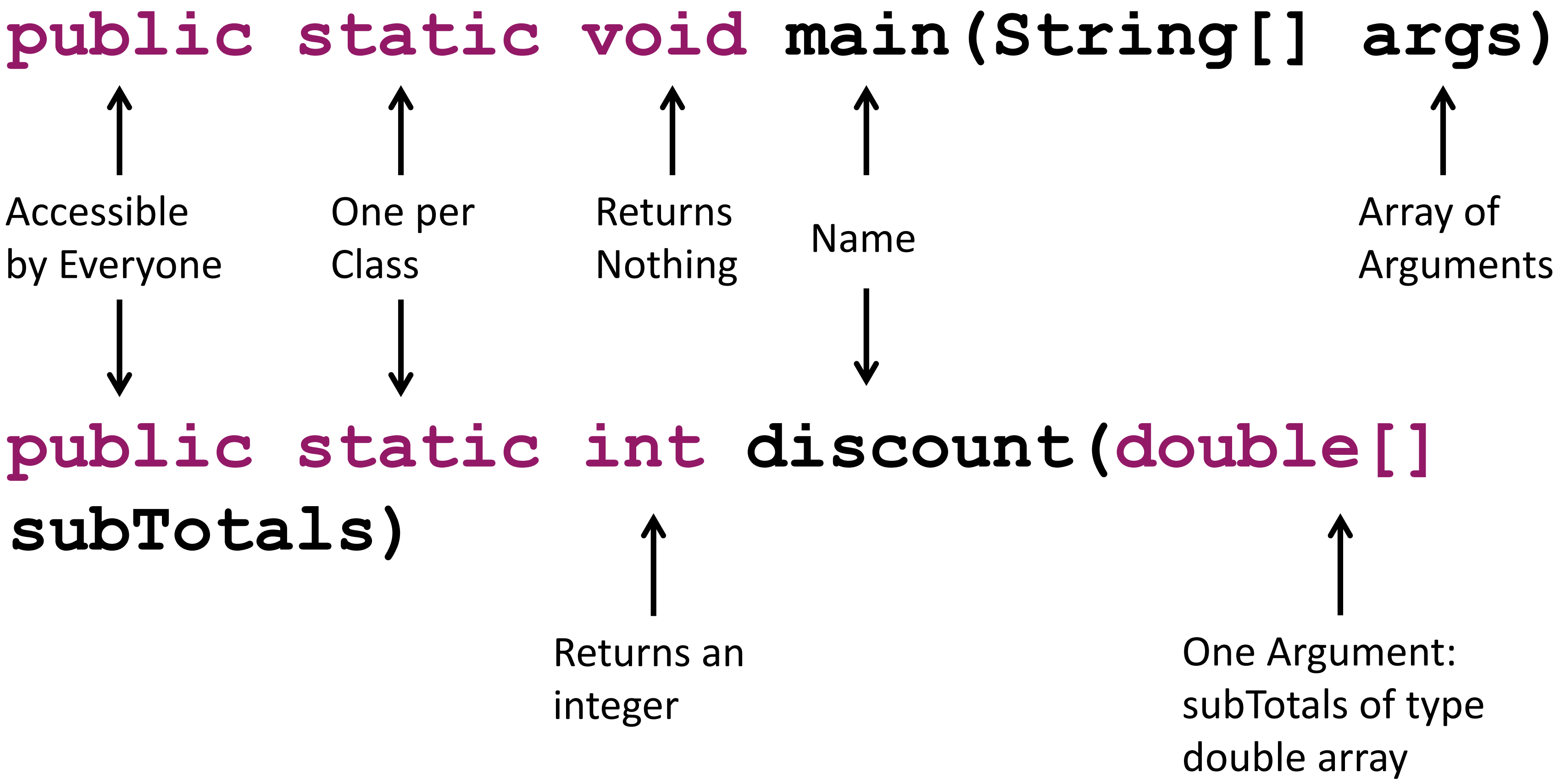
Array of Arguments

```
public static int read(Scanner in)
```

Returns an integer

One Argument of type Scanner

Methods



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

Accessible by Everyone

One per Class

Returns Nothing

Name

Array of Arguments

```
public static int discount(double[]
subTotals)
```

Returns an integer

One Argument: subTotals of type double array

Simple Example

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

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

OUTPUT:

Hi, my name is Daniel

↑

Flow of program

Compile Error

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

Sum Example

```
public class SumExample{
  // Method Declaration like variables (callee)
  public static int sum(int num1, int num2) {    #3
    System.out.println("Num1 is " + num1);       #4
    System.out.println("Num2 is " + num2);       #5
    int total = num1 + num2;                     #6
    return total;                                #7
  }

  public static void main(String[] args) {       #1
    int sum #8 = sum(13, 18);   // caller        #2
    System.out.println("Sum is " + sum);         #9
  }
}
```

Sum Example

```
public class SumExample{
  // Method Declaration like variables (callee)
  public static int sum(int num1, int num2) {          #3
    System.out.println("Num1 is " + num1);              #4
    System.out.println("Num2 is " + num2);              #5
    int total = num1 + num2;                            #6
    return total;                                       #7
  }

  public static void main(String[] args) {             #1
    int sum #8 = sum(13, 18);   // caller               #2
    System.out.println("Sum is " + sum);                #9
  }
}
```

Local variables
for sum only

Output:

```
    Num1 is 13
    Num2 is 18
    Sum is 31
```

Sum Usage

- Want to add 3 numbers (10, 13, 18)

- First Option
  - `int total1 = sum(13, 18);`
  - `int total = sum(10, total1);`
- Second Option (Substitution)
  - `int total = sum(10, sum(13,18));`
- Third Option (Commutative +)
  - `int total = sum(sum(13, 18), 10);`

Sum Example: Scope

```
public class SumExample{
  // Method Declaration like variables (callee)
  public static int sum(int num1, int num2) {                      #6
    System.out.println("Num1 is " + num1);                         #7
    System.out.println("Num2 is " + num2);                         #8
    int total = num1 + num2;                                       #9
    return total;                                                  #10
  }

  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 = sum(num2, num1); //(caller switched arguments) #5
     System.out.println("Sum is " + total);                        #12
  }
}
```

Sum Example: Scope

```
public class SumExample{
  // Method Declaration like variables (callee)
  public static int sum(int num1, int num2) {                     #6
    System.out.println("Num1 is " + num1);                        #7
    System.out.println("Num2 is " + num2);                        #8
    int total = num1 + num2;                                      #9
    return total;                                                 #10
  }

  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 = sum(num2, num1);  //(caller switched arguments) #5
     System.out.println("Sum is " + total);                       #12
  }
}
```

```
Output:  Main num1 is 18
         Main num2 is 13
         Num1 is 13
         Num2 is 18
         Sum is 31
```

Variables : Scope

```
// Method Declaration like variables (callee)
public static int sum(int num1, int num2) {
  System.out.println("Num1 is " + num1);
  System.out.println("Num2 is " + num2);
  int total = num1 + num2;
  num1 = 100;
  return total;
}

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

← No Effect : Logical Error

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