ICS 141
Programming with Objects

Jessica Maistrovich

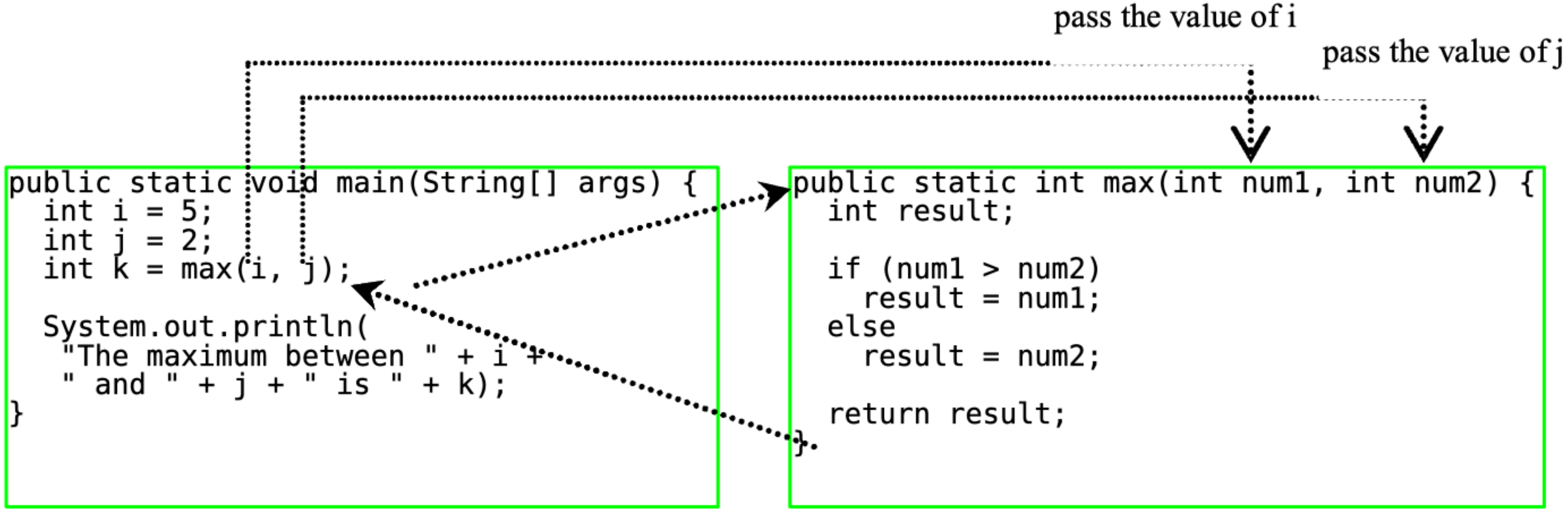
Metropolitan State University

Review sequential flow

Flow of program execution

- main() is the basic method in any Java Application
- Java Virtual Machine (JVM) starts the execution from the `main()` method
- When a method is called, control is transferred to the first statement in the body of the called method
- After the last statement of the called method is executed, control is passed back to the point immediately following the method call in `main()`

Slide provided by Thanaa M. Ghanem



```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem

i is now 5

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem

j is now 2

```
public static void main(Strin   args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

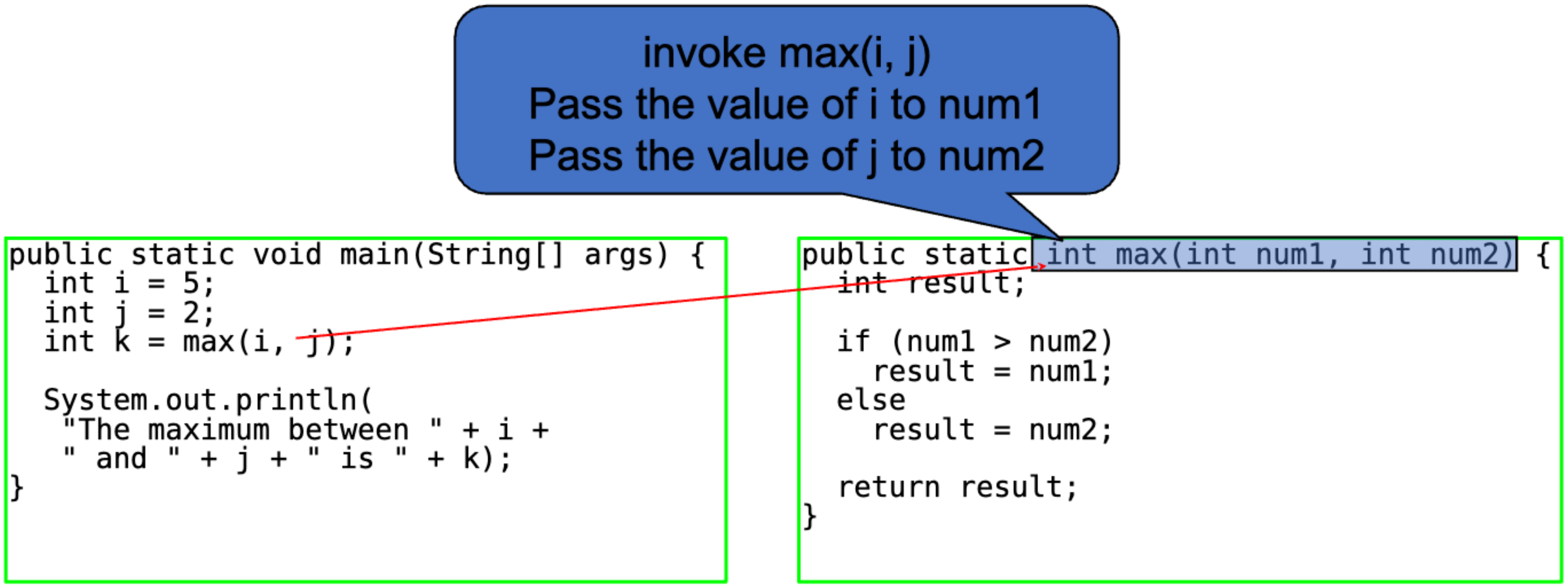
Slide provided by Thanaa M. Ghanem

invoke max(i, j)

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem



invoke max(i, j)
Pass the value of i to num1
Pass the value of j to num2

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

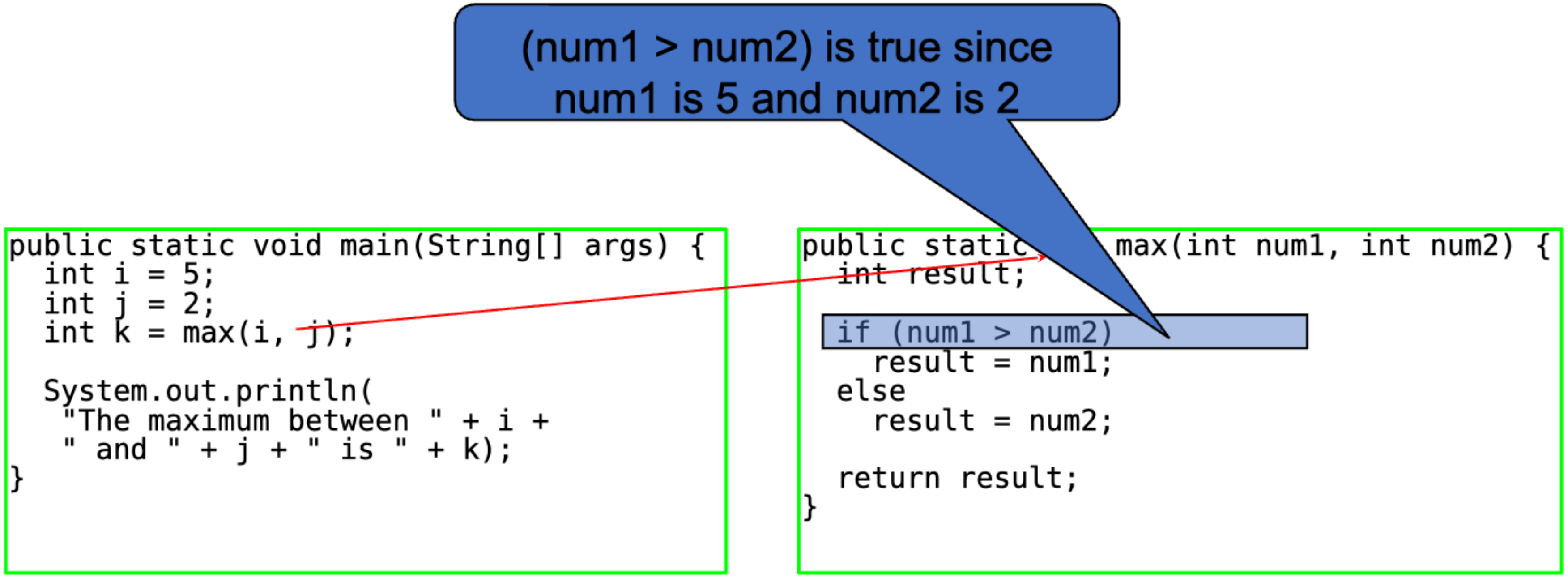
Slide provided by Thanaa M. Ghanem

declare variable result

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem

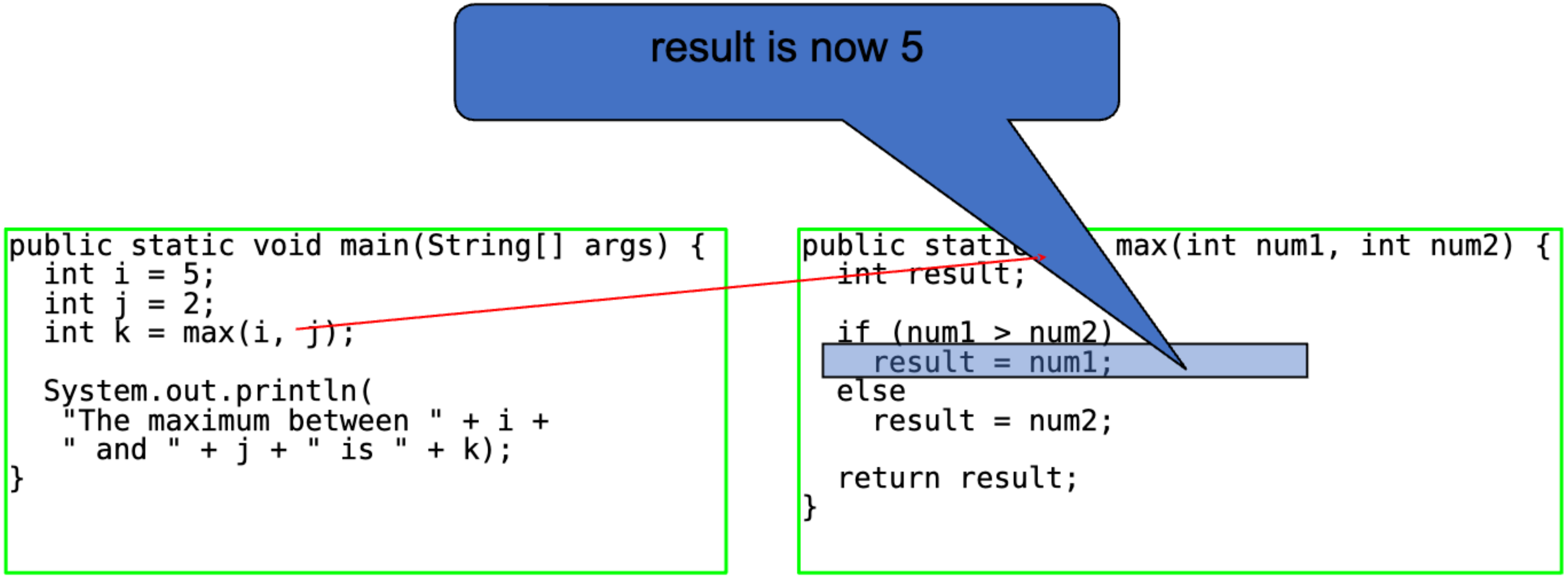


(num1 > num2) is true since num1 is 5 and num2 is 2

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static     max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem



result is now 5

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public stati     max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

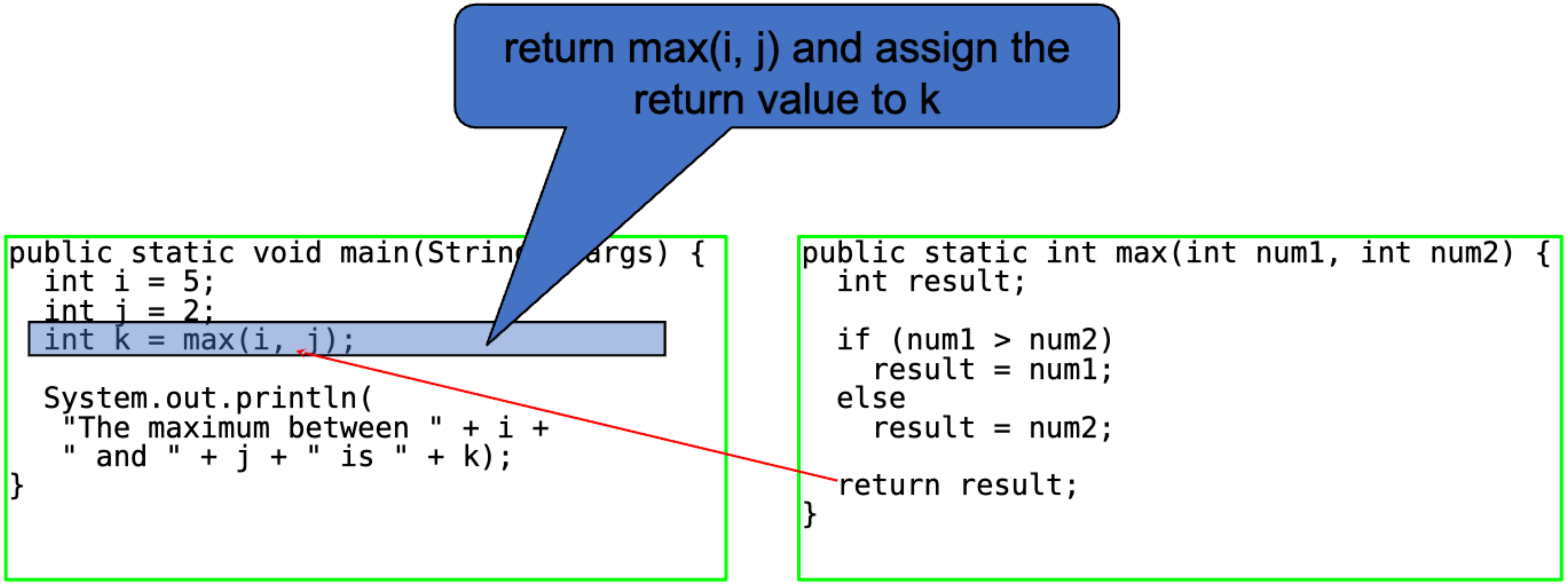
Slide provided by Thanaa M. Ghanem

return result, which is 5

```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
pu      tatic int max(int num1, int num2) {
       esult;

      num1 > num2)
     esult = num1;
    se
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem



return max(i, j) and assign the return value to k

```java
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```java
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Slide provided by Thanaa M. Ghanem

Execute the print statement

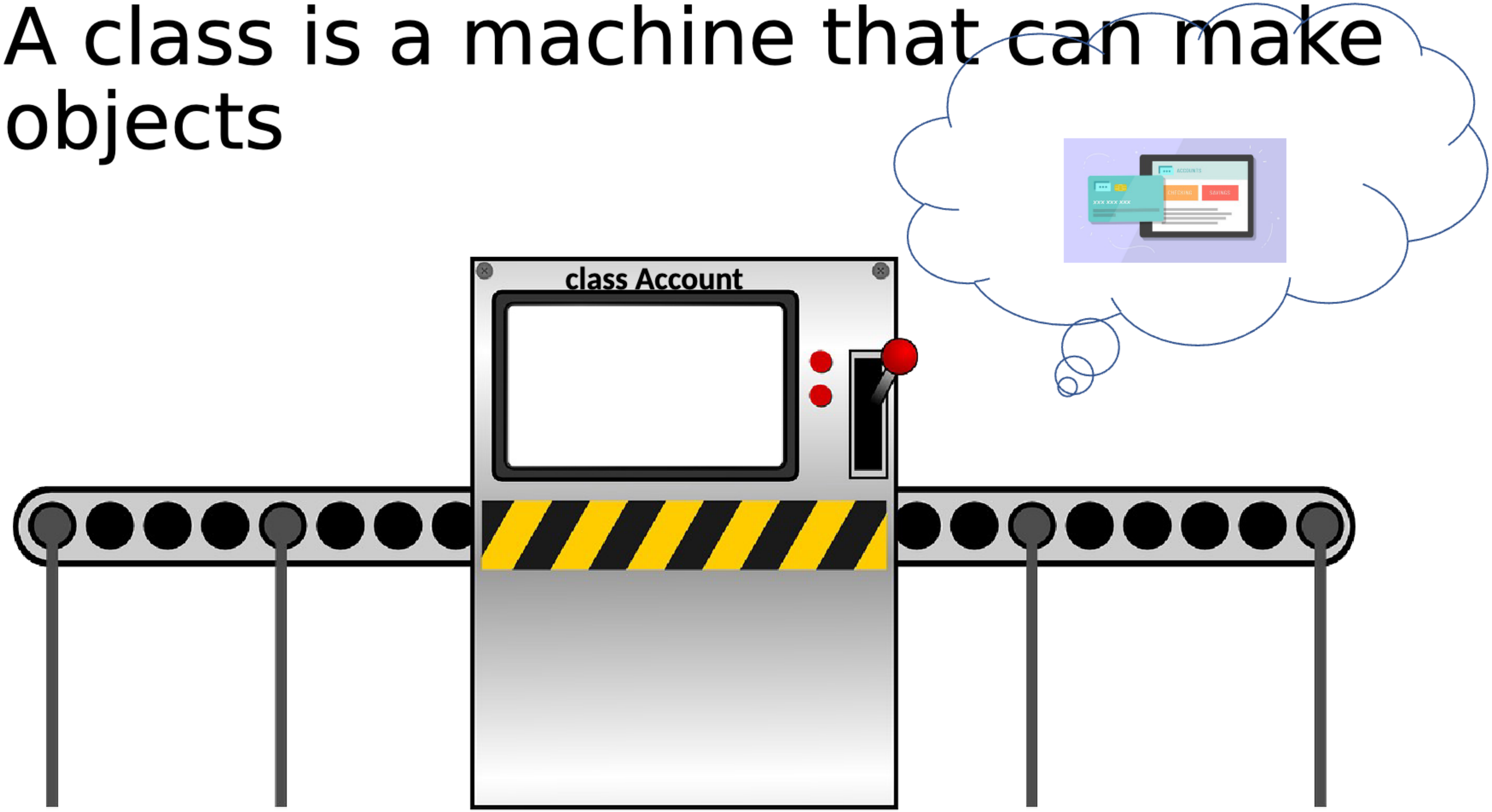
```
public static void main(String[] args) {
  int i = 5;
  int j = 2;
  int k = max(i, j);

  System.out.println(
   "The maximum between " + i +
   " and " + j + " is " + k);
}
```

```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

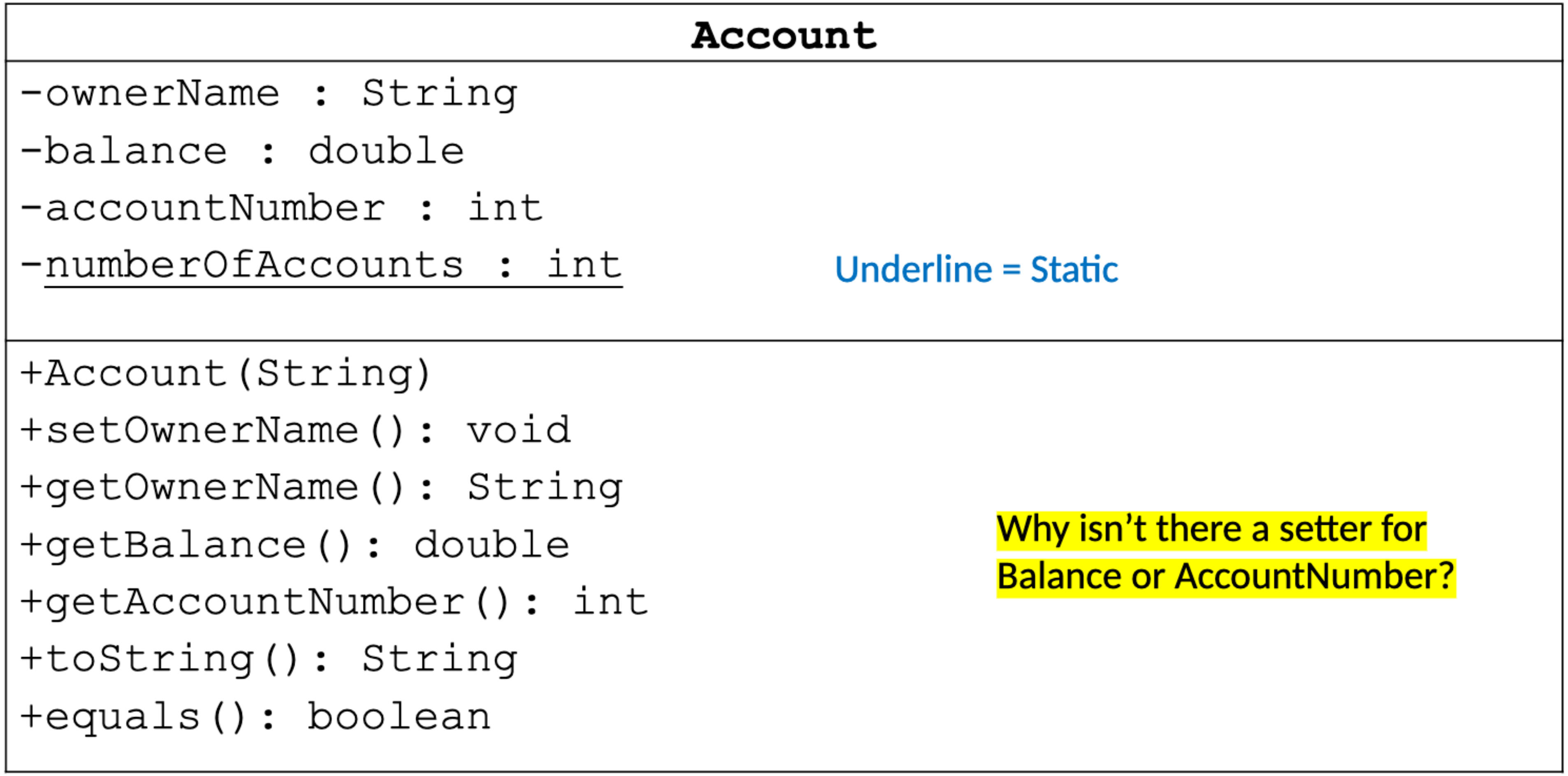
Slide provided by Thanaa M. Ghanem

Review creating classes

A class is a machine that can make objects



UML Diagram for `class Account`



```
Account
-ownerName : String
-balance : double
-accountNumber : int
-numberOfAccounts : int

+Account(String)
+setOwnerName(): void
+getOwnerName(): String
+getBalance(): double
+getAccountNumber(): int
+toString(): String
+equals(): boolean
```

Underline = Static

Why isn’t there a setter for Balance or AccountNumber?

Plus or Minus = Privacy

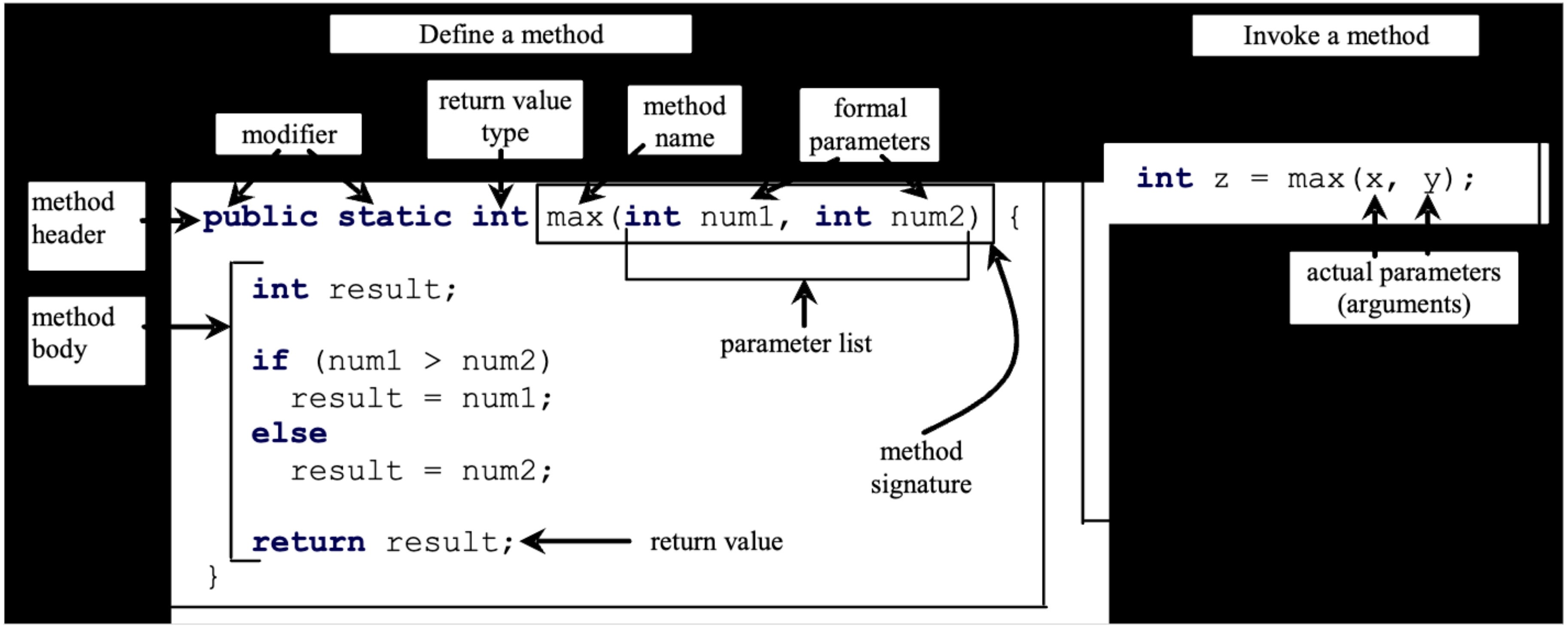
Try it! (part 1)

- Open Eclipse and create a new java project called AccountApplication.
- Create a driver class called AccountDriver. Include the main method.
- In the main method, declare and instantiate a Scanner object.
- Collect the name from the user and store it in a variable.
- Print the name to verify that you have correctly collected it.
- Create a class called account based on the UML in previous slide.
    - The constructor should accept one parameter (the name of the account holder). The balance should be initialized to zero. The account number is determined by using the static variable.
- In the driver class, declare and instantiate an Account object. Use the name collected from the user as the name of the account holder.
- Print the account.

User defined methods

If you have opened an account what do you expect to be able to do with it?

Defining Methods



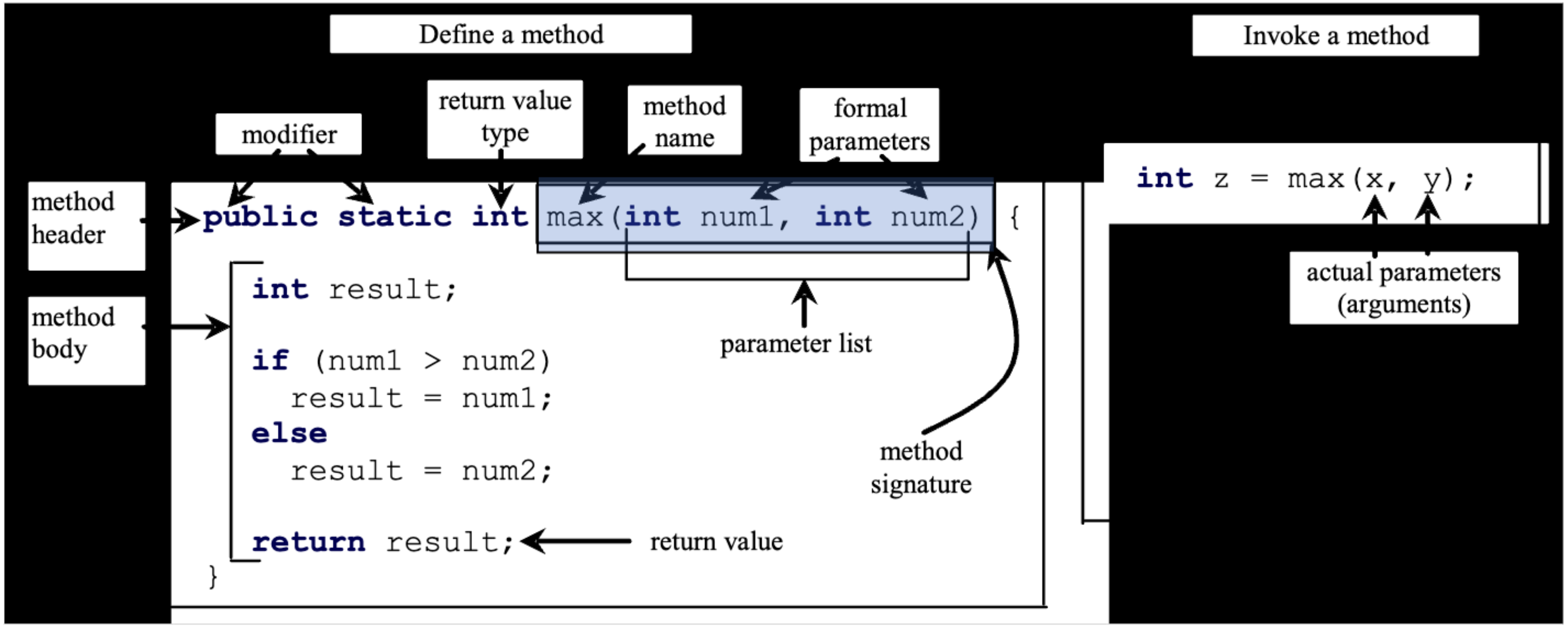
```
public static int max(int num1, int num2) {

  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

```
int z = max(x, y);
```

Slide provided by Thanaa M. Ghanem

Method Signature

Method signature is the combination of the method name and the parameter list.



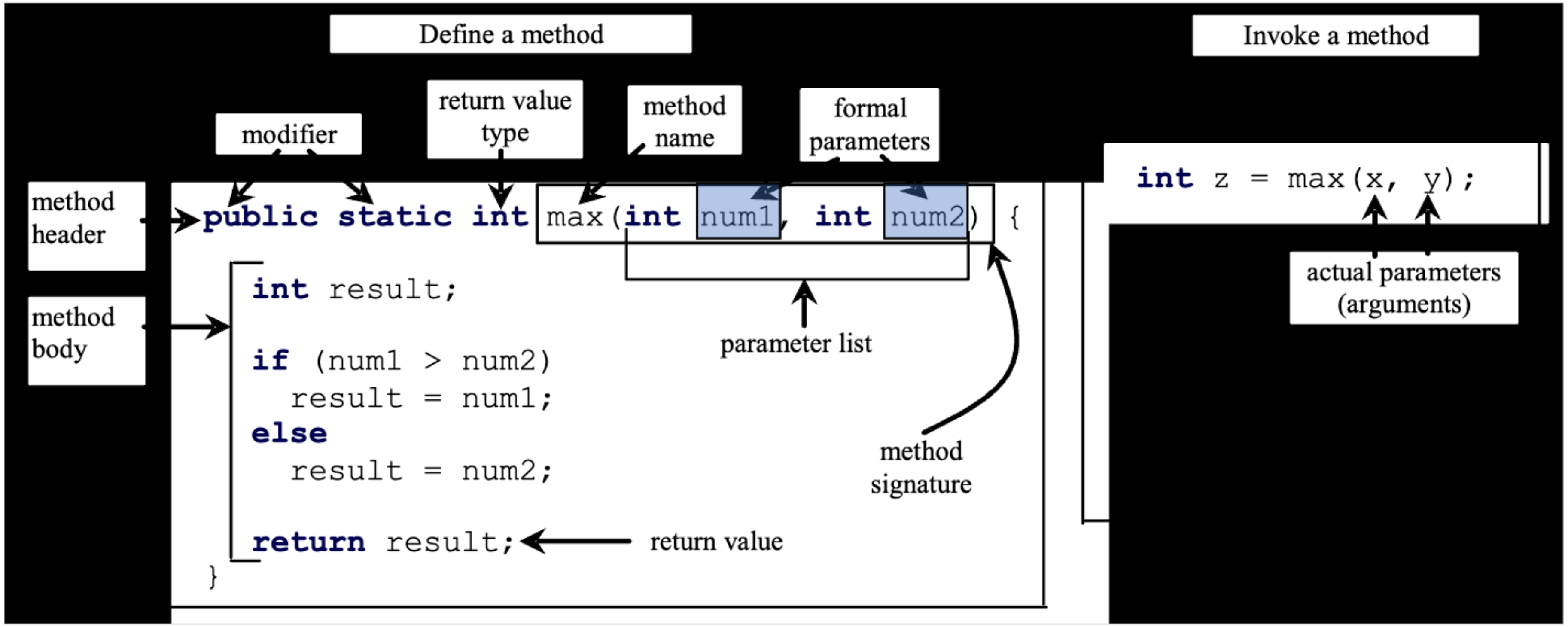
```
public static int max(int num1, int num2) {

  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

```
int z = max(x, y);
```

Slide provided by Thanaa M. Ghanem

Formal Parameters

The variables defined in the method header are known as *formal parameters*.



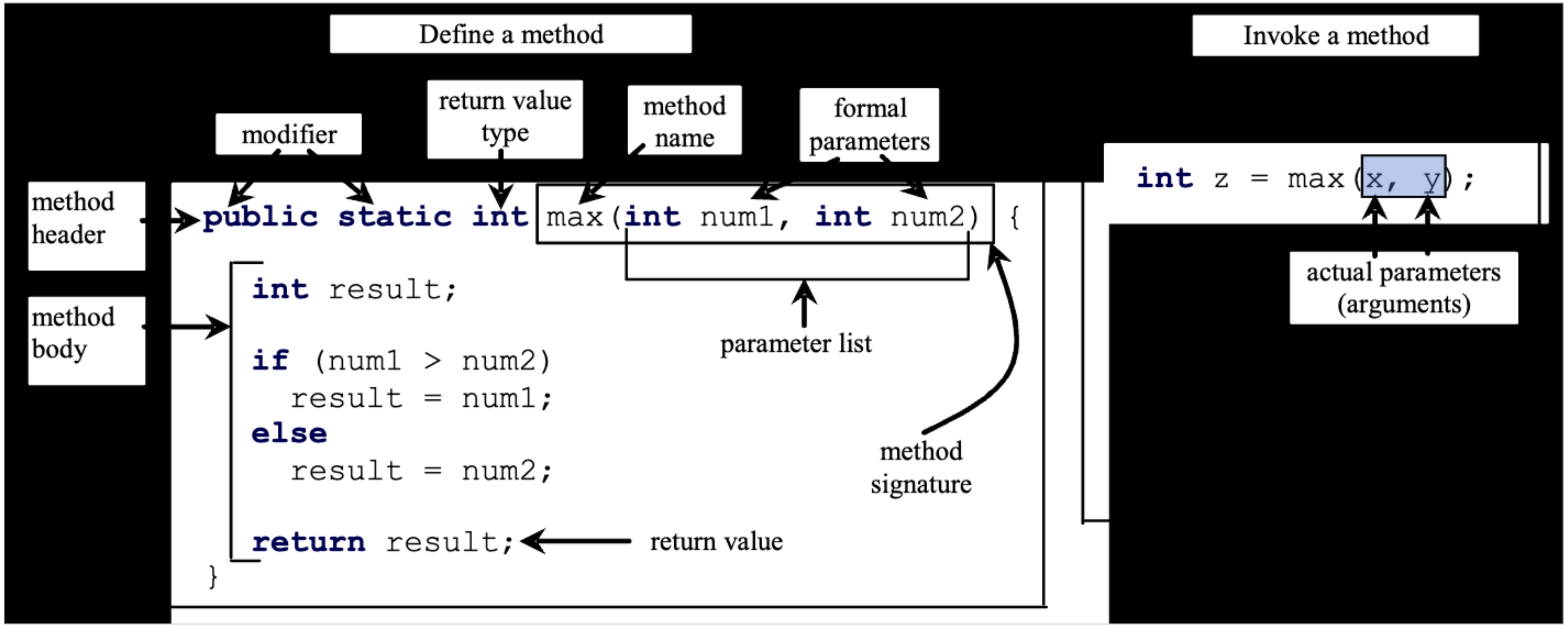
```
public static int max(int num1, int num2) {

   int result;

   if (num1 > num2)
     result = num1;
   else
     result = num2;

   return result;
}
```

```
int z = max(x, y);
```

Slide provided by Thanaa M. Ghanem

Actual Parameters or arguments

When a method is invoked, you pass a value to the parameter. This value is referred to as *actual parameter or argument*.



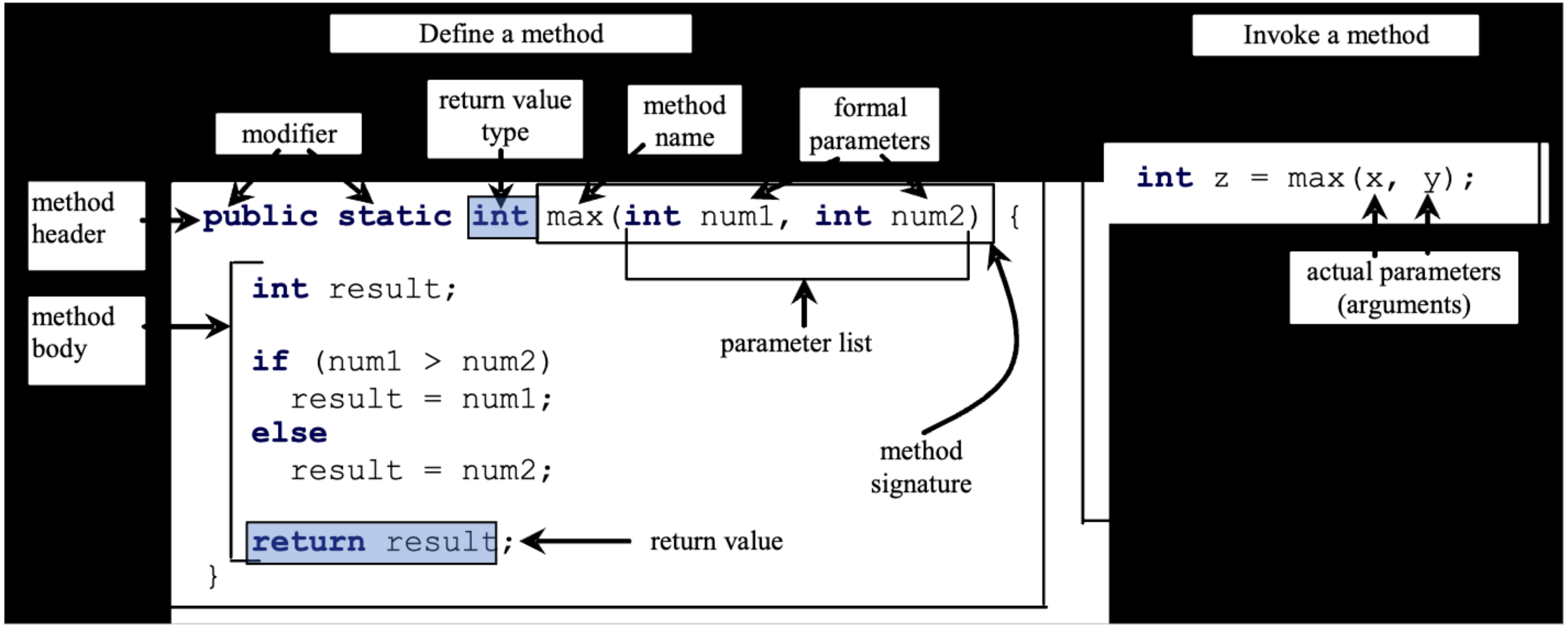
```
public static int max(int num1, int num2) {

    int result;

    if (num1 > num2)
      result = num1;
    else
      result = num2;

    return result;
}
```

```
int z = max(x, y);
```

Slide provided by Thanaa M. Ghanem

Return Value Type

A method may return a value. The `returnValueType` is the data type of the value the method returns. If the method does not return a value, the `returnValueType` is the keyword `void`. For example, the returnValueType in the `main` method is `void`.



```
public static int max(int num1, int num2) {
    int result;

    if (num1 > num2)
      result = num1;
    else
      result = num2;

    return result;
}
```

```
int z = max(x, y);
```

Slide provided by Thanaa M. Ghanem

Choosing Method Name

- Each method should be limited to performing a single, well-defined task

- A method name should be concise name that expresses what the function does

- If you cannot choose a concise name then most probably your method is attempting to perform too many diverse tasks. It is usually best to break such a method into several smaller methods.

- Choosing meaningful method names and meaningful parameter names makes programs more readable and helps avoid excessive use of comments

Slide provided by Thanaa M. Ghanem

Math Symbols in Java

+, -, *, /, %

Note - division will automatically be integer division if both dividend and divisor are integers, otherwise it will be real division.

For exponents – have to use math class

Math.pow(10, 2) is 100

Can use shortcuts +=, -=, ++, etc.

Chapter 10 in book discusses in depth.

Try it!

- Create a method in the Account class to deposit funds into the account. What will you name it? Are there any inputs? Are there any outputs?
- Create a method in the Account class to withdraw funds from the account. What will you name it? Are there any inputs? Are there any outputs?

- In the driver class, deposit $1000 into the account you made previously. Print the account. (Do not delete the previous printing, this is how we check to see the method is working). How will you know the deposit method is working?
- In the driver class, withdraw $500 from the account you made previously. Print the account. How will you know the withdraw method is working?