Some terminology

$$\begin{tikzcd}
 & & \text{private} & \\
 & \text{attribute} \arrow[ur] \arrow[dr] & & \\
 & & \text{public} \arrow[dr] & \\
\text{feature} \arrow[uur] \arrow[ddr] & & & \text{interface} \\
 & & \text{public} \arrow[ur] & \\
 & \text{method} \arrow[ur] \arrow[dr] & & \\
 & & \text{private} &
\end{tikzcd}$$

public attribute = field[1]

[1] Not everyone uses this convention. Some use attribute and field as synonyms.

More terminology

Consider the API of the class `Currency`. It contains the method

```
public static double convert(double amount,
   String from, String to)
```

This method has three parameters named `amount`, `from` and `to`.

[2] The textbook calls these parameters as well. On a test, you may call them either arguments or parameters.

More terminology

Consider the API of the class `Currency`. It contains the method

```
public static double convert(double amount,
   String from, String to)
```

This method has three parameters named `amount`, `from` and `to`.

Consider the following statement.

```
double priceInCAD = Currency.convert(priceInUSD,
   Currency.USD, Currency.CAD);
```

This method invocation takes three arguments,[2] namely `priceInUSD`, `Currency.USD` and `Currency.CAD`.

[2] The textbook calls these parameters as well. On a test, you may call them either arguments or parameters.

Screen output

Question

How do you print the price of gold on the screen?

Screen output

Question

How do you print the price of gold on the screen?

Answer

```
System.out.printf("$ %.2f\n", price);
```

Screen output

Question

How do you print the price of gold on the screen?

Answer

```
System.out.printf("$ %.2f\n", price);
```

Question

- `System` is a

Screen output

Question

How do you print the price of gold on the screen?

Answer

```
System.out.printf("$ %.2f\n", price);
```

Question

- System is a class.
- out is an

Screen output

Question

How do you print the price of gold on the screen?

Answer

```
System.out.printf("$ %.2f\n", price);
```

Question

- `System` is a class.
- `out` is an attribute.
- `printf` is a

Screen output

Question

How do you print the price of gold on the screen?

Answer

```
System.out.printf("$ %.2f\n", price);
```

Question

- `System` is a class.
- `out` is an attribute.
- `printf` is a method.

Screen output

Question

How can we determine the type of the attribute `System.out`?

Screen output

Question

How can we determine the type of the attribute `System.out`?

Answer

Study the API of the `System` class.

Screen output

Question

How can we determine the type of the attribute `System.out`?

Answer

Study the API of the `System` class.

The type of `System.out` is `PrintStream`.

Screen output

```
import java.io.PrintStream;
...
   PrintStream output = System.out;
   output.printf("$ %.2f\n", price);
```

Screen output

Question

What is the signature of the `println` method in

```
output.println("It is Monday!");
```

Screen output

Question

What is the signature of the `println` method in

```
output.println("It is Monday!");
```

Answer

`println(String)`.

Screen output

Question

What is the signature of the `println` method in

```
output.println(123);
```

Screen output

Question

What is the signature of the `println` method in

```
output.println(123);
```

Answer

`println(int)`.

Screen output

Question

What is the signature of the `println` method in

```
boolean isSunny = false;
output.println(isSunny);
```

Screen output

Question

What is the signature of the `println` method in

```
boolean isSunny = false;
output.println(isSunny);
```

Answer

`println(boolean)`.

Screen output

Question

What is the signature of the `println` method in

```
output.println('\u226E');
```

Screen output

Question

What is the signature of the `println` method in

```
output.println('\u226E');
```

Answer

`println(char)`.

Screen output

Question

What is the signature of the `println` method in

```
output.println();
```

Screen output

Question

What is the signature of the `println` method in

```
output.println();
```

Answer

`println()`.

Keyboard input

Question

How do you get the amount of gold from the keyboard?

Keyboard input

Question

How do you get the amount of gold from the keyboard?

```
import java.util.Scanner;
...
   Scanner input = new Scanner(System.in);
```

Next week we will discuss what `new Scanner` does.

Keyboard input

Question

What is the return type of the `nextInt` method in

```
input.nextInt();
```

Keyboard input

Question

What is the return type of the `nextInt` method in

```
input.nextInt();
```

Answer

`int`.

Of course, the result should saved in a variable.

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

Keyboard input

Question

What is the return type of the `next` method in

```
input.next();
```

Keyboard input

Question

What is the return type of the `next` method in

```
input.next();
```

Answer

`String`.

Of course, the result should saved in a variable.

```
String token = input.next();
```

Keyboard input

Question

What is the return type of the `nextLine` method in

```
input.nextLine();
```

Keyboard input

Question

What is the return type of the `nextLine` method in

```
input.nextLine();
```

Answer

`String`.

Of course, the result should saved in a variable.

```
String line = input.nextLine();
```

The price of gold

Write an app that prompts the user "Enter the amount of gold in kilos: " and, after the user has entered the amount k, prints on the screen "The Price of k kilos of Gold " followed by the current price of k kilos of Gold in Canadian dollars. If the users enters a negative amount, the app crashes with the message "The amount of gold cannot be negative."

Template

```
import java.io.PrintStream;
import java.util.Scanner;

public class
{
   public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);
      PrintStream output = System.out;

   }
}
```

How to use Java on your own computer?

Instructions can be found at the URL
`www.cse.yorku.ca/~roumani/jba/lab2`
As editor, I suggest jEdit which can be downloaded from the URL
`www.jedit.org`
Once you are familiar with jEdit, you may try eclipse which can be downloaded from the URL
`www.eclipse.org`

Path

PATH is an environment variable that specifies a list of directories where executable programs are located.

To use the programs java, javac and jedit, the directories, in which the executable programs javac.exe, java.exe and jedit.exe can be found, should be part of PATH.

To see this list of directories, type in the command prompt PATH.[3]

To set PATH, do a web search for how to set an environment variable in Windows.

[3] or path or Path or patH, etc.

Classpath

CLASSPATH is an environment variable that specifies a list of directories and jar files that contain Java bytecode.

To use, for example, the `Gold` class of the package `franck.cse1020`, which is stored in the jar file
`www.eecs.yorku.ca/course_archive/2013-14/F/classpath/1020/franck.jar`
save the jar file franck.jar and ensure that it is part of the CLASSPATH.

To see this list of directories and jarfiles, type in the command prompt echo %CLASSPATH%.

To set CLASSPATH, do a web search for how to set an environment variable in Windows. See also Section 2.2.4 of the textbook for an alternative way to handle jar files.

Memory model

Running an app results in invoking its main method.

When a method is invoked, a block of memory is allocated to store the values of the parameters and variables of the method.

Main method

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

Question

How many parameters does the `main` method have?

[4] We will come back to this type later in the course.

Main method

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

Question

How many parameters does the `main` method have?

Answer: one.

[4] We will come back to this type later in the course.

Main method

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

Question

How many parameters does the `main` method have?

Answer: one.

Question

What is the name of the parameter?

[4] We will come back to this type later in the course.

Main method

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

Question

How many parameters does the main method have?

Answer: one.

Question

What is the name of the parameter?

Answer: `args`.

[4] We will come back to this type later in the course.

Main method

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

Question

How many parameters does the `main` method have?

Answer: one.

Question

What is the name of the parameter?

Answer: `args`.

Question

What is the type of the parameter?

[4] We will come back to this type later in the course.

Main method

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

Question

How many parameters does the `main` method have?

Answer: one.

Question

What is the name of the parameter?

Answer: `args`.

Question

What is the type of the parameter?

Answer: `String[]`.[4]

[4] We will come back to this type later in the course.

Main method

In the first half of this course, we will not use the parameter of the main method. Therefore, we will not include the parameter of the main method in our memory diagrams (for now).

Price of gold

Simplified version of body of the main method:

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
   GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
   Currency.convert(priceInUSD, Currency.USD, Currency.CAD
```

Question

What are the names of the variables in the above main method?

Price of gold

Simplified version of body of the main method:

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
   GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
   Currency.convert(priceInUSD, Currency.USD, Currency.CAD
```

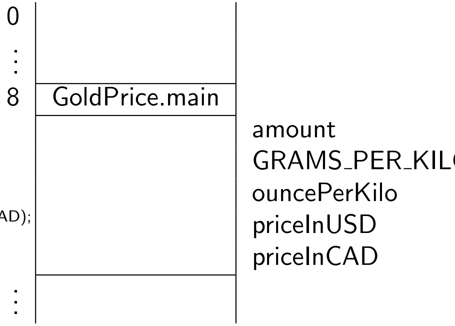
Question

What are the names of the variables in the above main method?

Answer: `amount`, `GRAMS_PER_KILO`, `ouncePerKilo`, `priceInUSD` and `priceInCAD`.

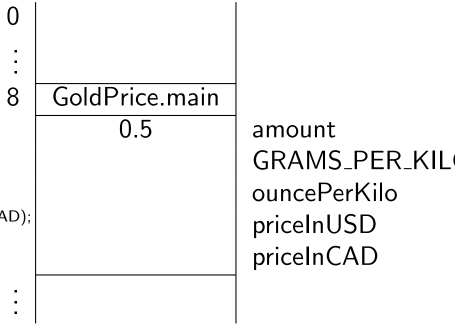
Memory model

```
double amount = 0.5;
 final  double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



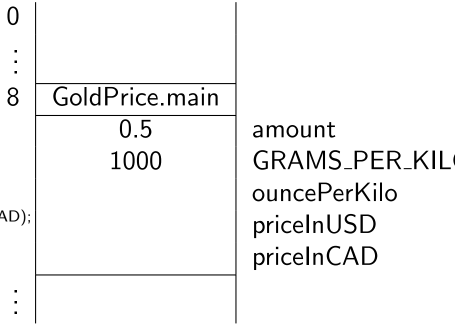
Memory model

```
double amount = 0.5;
 final  double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
   GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
   Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



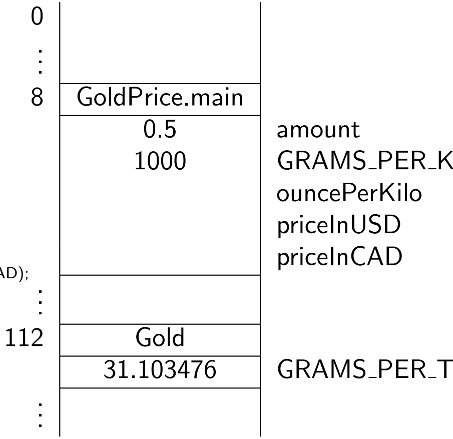
Memory model

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



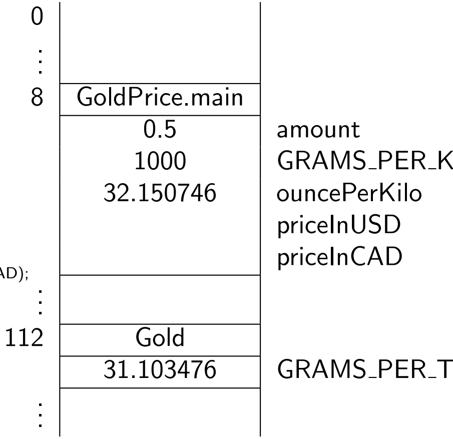
Memory model

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



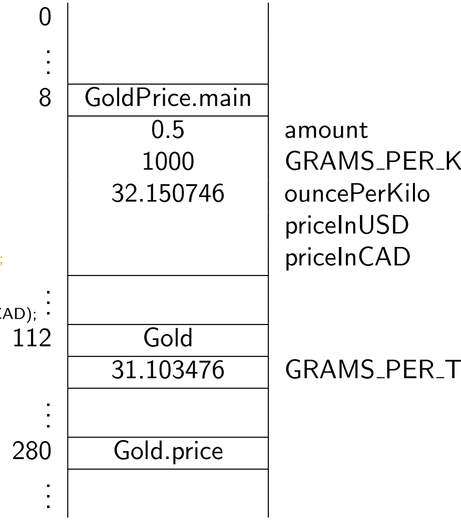
Memory model

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



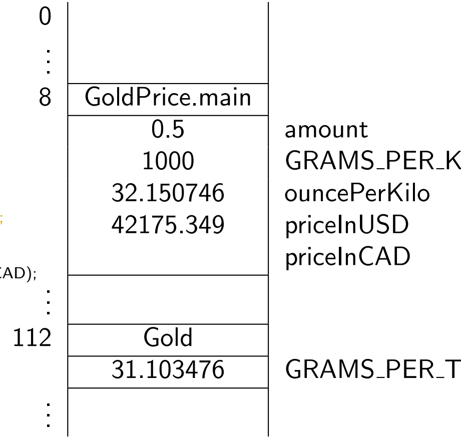
Memory model

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



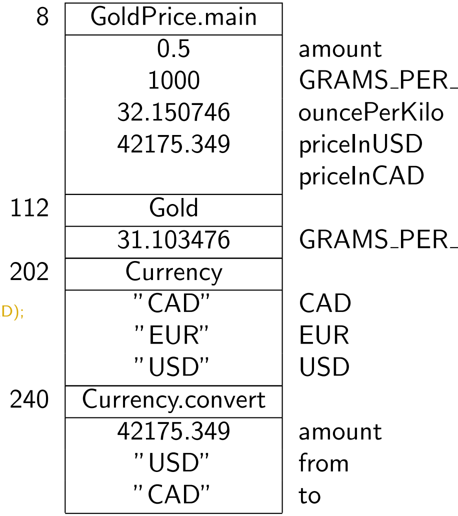
Memory model

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



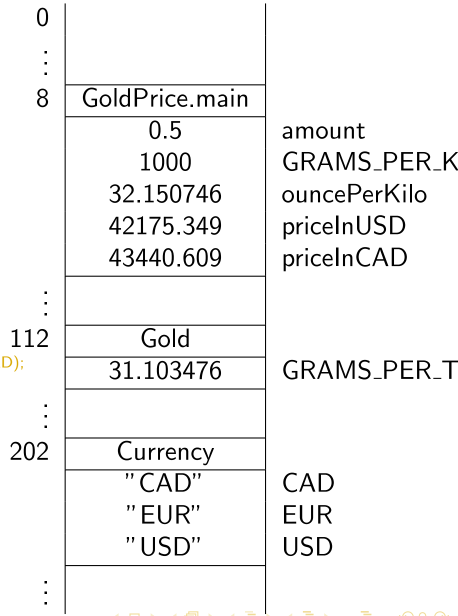
Memory model

```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



Memory model

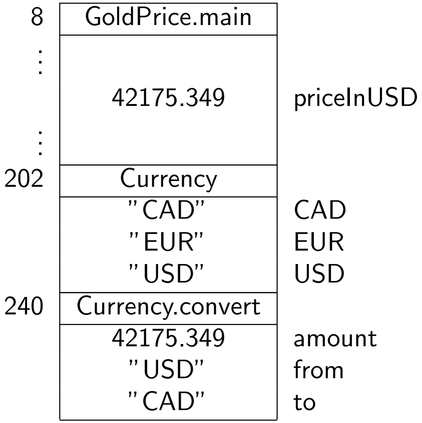
```
double amount = 0.5;
final double GRAMS_PER_KILO = 1000;
double ouncePerKilo =
    GRAMS_PER_KILO / Gold.GRAMS_PER_TROY_OUNCE;
double priceInUSD = amount * ouncePerKilo * Gold.price();
double priceInCAD =
    Currency.convert(priceInUSD, Currency.USD, Currency.CAD);
```



Pass-by-value

... Currency.convert(priceInUSD, Currency.USD, Currency.CAD);

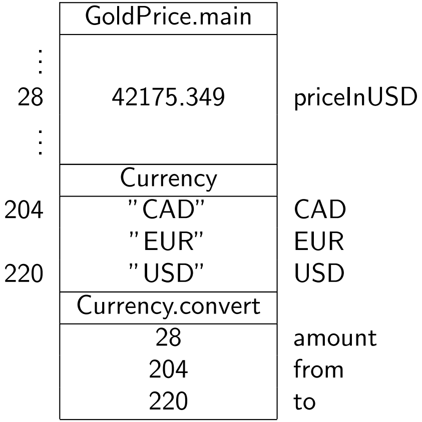
The values of the arguments are passed in Java (and many other programming languages).



Pass-by-reference

... Currency.convert(priceInUSD, Currency.USD, Currency.CAD);

The addresses of the arguments are passed in some programming languages such as Perl.



Pass-by-value

Question

```
int value = 2;
Magic.triple(value);
output.println(value);
```

What is the output produced by the above snippet?

Pass-by-value

Question

```
int value = 2;
Magic.triple(value);
output.println(value);
```

What is the output produced by the above snippet?

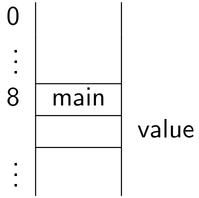
Answer

2

The method triple of the class Magic gets passed only the value of the variable value, not its address.

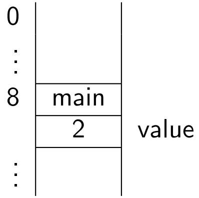
Pass-by-value

```
int value = 2;
Magic.triple(value);
output.println(value);
```



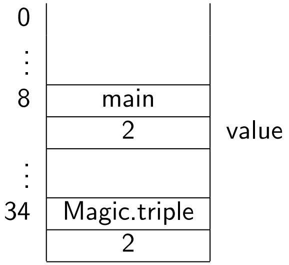
Pass-by-value

```
int value = 2;
Magic.triple(value);
output.println(value);
```



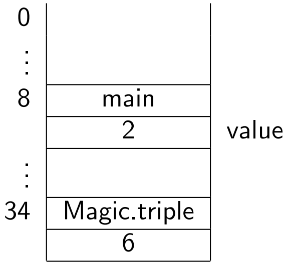
Pass-by-value

```
int value = 2;
Magic.triple(value);
output.println(value);
```



Pass-by-value

```
int value = 2;
Magic.triple(value);
output.println(value);
```



Pass-by-reference

Question

```
int value = 2;
Magic.triple(value);
output.println(value);
```

If Java were to use pass-by-reference, what would the output produced by the above snippet be?

Pass-by-reference

Question

```
int value = 2;
Magic.triple(value);
output.println(value);
```

If Java were to use pass-by-reference, what would the output produced by the above snippet be?

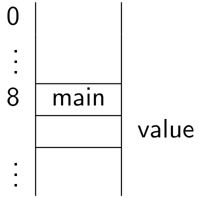
Answer

6 or any other integer.

The method triple of the class Magic gets passed the address of the variable value and, hence, can change its value.

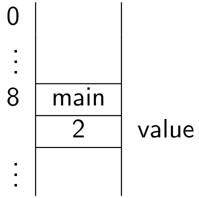
Pass-by-reference

```
int value = 2;
Magic.triple(value);
output.println(value);
```



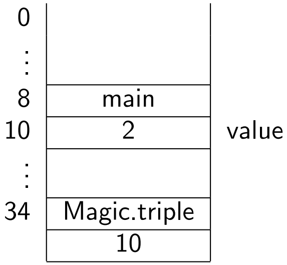
Pass-by-reference

```
int value = 2;
Magic.triple(value);
output.println(value);
```



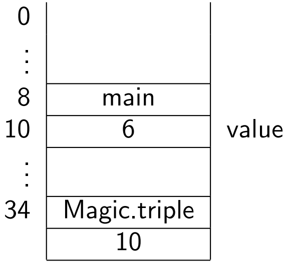
Pass-by-reference

```
int value = 2;
Magic.triple(value);
output.println(value);
```



Pass-by-value

```
int value = 2;
Magic.triple(value);
output.println(value);
```



Overloading

The signature of a method is unique in its class.

Terminology

Two methods in the same class with the same name are said to be overloaded.

Example

In the class PrintStream, the method println is overloaded.

Early binding

When the compiler encounters the invocation

$\mathrm{C.m}(a_1, \ldots, a_n)$

it must determine which method to invoke. This process is known as early binding. It consists of the following three steps.

1. Find the class C.
2. Find a compatible method m in class C.
3. Select the most specific compatible method m in class C.

Early binding (step 1)

Early binding of $\mathrm{C}.\mathrm{m}(a_1, \ldots, a_n)$.

Question

How can "find the class C" fail?

Early binding (step 1)

Early binding of $\mathrm{C}.\mathrm{m}(a_1, \ldots, a_n)$.

Question

How can "find the class C" fail?

Answer

The class is missing, since it has not been imported, it is not part of the classpath, or its name has been misspelled.

Early binding (step 2)

Early binding of $\mathrm{C.m}(a_1, \ldots, a_n)$.

Question

When is a method m in class C compatible with invocation $\mathrm{C.m}(a_1, \ldots, a_n)$?

Answer

The types of the arguments $a_1, \ldots, a_n$ are compatible with the types of the parameters of the method m.

Early binding (step 2)

Question

Which methods in class $\mathrm{PrintStream}$ are compatible with invocation $\mathrm{output.println}(1)$?

Early binding (step 2)

Question

Which methods in class PrintStream are compatible with invocation output.println(1)?

Answer

println(double)
println(float)
println(int)
println(long)

Early binding (step 2)

Early binding of $\mathrm{C}.\mathrm{m}(a_1, \ldots, a_n)$.

Question

How can "find a compatible method m in class C" fail?

Early binding (step 2)

Early binding of $\mathrm{C.m}(a_1, \ldots, a_n)$.

Question

How can "find a compatible method m in class C" fail?

Answer

The method is missing, since it simply does not exist or its name has been misspelled.

Early binding (step 3)

Question

Which of the methods

$\text{println}\,(\text{double})$
$\text{println}\,(\text{float})$
$\text{println}\,(\text{int})$
$\text{println}\,(\text{long})$

in class PrintStream is most specific to invocation $\text{output.println}\,(1)$?

Early binding (step 3)

Question

Which of the methods

$$\text{println}(\text{double})$$
$$\text{println}(\text{float})$$
$$\text{println}(\text{int})$$
$$\text{println}(\text{long})$$

in class PrintStream is most specific to invocation output.println(1)?

Answer

println(int) since the argument 1 is of type int.

Early binding (step 3)

Question

Which of the methods

$\text{println}\,(\text{double})$
$\text{println}\,(\text{float})$
$\text{println}\,(\text{int})$
$\text{println}\,(\text{long})$

in class PrintStream is most specific to invocation $\text{output.println}(1\text{L})$?

Early binding (step 3)

Question

Which of the methods

println(double)
println(float)
println(int)
println(long)

in class PrintStream is most specific to invocation output.println(1L)?

Answer

println(long) since the argument 1L is of type long.

Early binding (step 3)

Question

Which of the methods

println (double)
println (float)
println (int)
println (long)

in class PrintStream is most specific to invocation
output.println ('1') ?

Early binding (step 3)

Question

Which of the methods

println (double)
println (float)
println (int)
println (long)

in class PrintStream is most specific to invocation output.println ('1') ?

Answer

println (int) since the argument '1' is of type char and converting it to an int requires the least amount of promotion.

Early binding (step 3)

Early binding of $\mathrm{C.m}(a_1, \ldots, a_n)$.

Question

How can "select the most specific compatible method m in class C" fail?

Early binding (step 3)

Early binding of $\mathrm{C.m}(a_1, \ldots, a_n)$.

Question

How can "select the most specific compatible method m in class C" fail?

Answer

Consider the class C with methods

$\mathrm{m(int,\ double)}$
$\mathrm{m(double,\ int)}$

and the invocation $\mathrm{C.m(1,\ 2)}$. Note that both $\mathrm{m(int,\ double)}$ and $\mathrm{m(double,\ int)}$ are compatible with $\mathrm{C.m(1,\ 2)}$. However, both require the same amount of promotion, namely promoting an int to a double. Hence, one is not more specific than the other and therefore we cannot select the most specific one.

To do

- Study Section 3.1 and 3.3 of the textbook.