

# The price of gold

## Problem

Write an app that prints the price of one kilo of gold in Canadian dollars.

# Static methods

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

## Question

What is the return type of the method `convert`?

# Static methods

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

## Question

What is the return type of the method `convert`?

## Answer

`double`.

# Static methods

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

## Question

What is the return type of the method `convert`?

## Answer

`double`.

## Question

How many parameters does the method `convert` have?

# Static methods

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

Question

What is the return type of the method `convert`?

Answer

`double`.

Question

How many parameters does the method `convert` have?

Answer

Three.

# Static methods

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

## Question

What is the signature of the method `convert`?

# Static methods

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

## Question

What is the signature of the method `convert`?

## Answer

```
convert(double, String, String)
```

## The precondition

```
amount >= 0.0,  
from == Currency.CAD || from == Currency.USD || from == Currency.EUR,  
to == Currency.CAD || to == Currency.USD || to == Currency.EUR
```

can be read as

amount  $\geq$  0.0 and

(from == Currency.CAD or from == Currency.USD or from == Currency.EUR) and  
(to == Currency.CAD or to == Currency.USD or to == Currency.EUR)



## The precondition

```
amount >= 0.0,  
from == Currency.CAD || from == Currency.USD || from == Currency.EUR,  
to == Currency.CAD || to == Currency.USD || to == Currency.EUR
```

can be read as

amount  $\geq$  0.0 and

(from == Currency.CAD or from == Currency.USD or from == Currency.EUR) and  
(to == Currency.CAD or to == Currency.USD or to == Currency.EUR)

### Question

Who is responsible for the precondition, the client of the implementer?

## The precondition

```
amount >= 0.0,  
from == Currency.CAD || from == Currency.USD || from == Currency.EUR,  
to == Currency.CAD || to == Currency.USD || to == Currency.EUR
```

can be read as

amount  $\geq$  0.0 and

(from == Currency.CAD or from == Currency.USD or from == Currency.EUR) and  
(to == Currency.CAD or to == Currency.USD or to == Currency.EUR)

### Question

Who is responsible for the precondition, the client of the implementer?

### Answer

The client.

## Question

If `Currency.convert(1.0, Currency.USD, Currency.CAD)` returns `-1.03`, who is to blame?

## Question

If `Currency.convert(1.0, Currency.USD, Currency.CAD)` returns  $-1.03$ , who is to blame?

## Answer

The implementer, since the client has done its job by providing arguments that satisfy the precondition, whereas the implementer did not satisfy the postcondition.

## Question

If `Currency.convert(1.0, Currency.USD, "YEN")` returns `-99.13`, who is to blame?

## Question

If `Currency.convert(1.0, Currency.USD, "YEN")` returns `-99.13`, who is to blame?

## Answer

The client, since the third argument "YEN" does not satisfy the precondition, which is the client's responsibility.

## Problem

Write an app that pops up a dialog box with the title “The Price of Gold” and the message “Enter the amount of gold in kilos” and, after the user has entered the amount  $k$  and clicks the OK button, pops up another dialog box with the title “The Price of  $k$  kilos of Gold” and the current price of  $k$  kilos of Gold in Canadian dollars.

# Static methods

```
public static String showInputDialog(Component  
parent, String message, String title, int  
messageType)1 of class JOptionPane of package javax.swing.
```

## Question

What is the return type of the method `showInputDialog`?

---

<sup>1</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.



# Static methods

```
public static String showInputDialog(Component  
parent, String message, String title, int  
messageType)1 of class JOptionPane of package javax.swing.
```

## Question

What is the return type of the method `showInputDialog`?

## Answer

String.

---

<sup>1</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

```
public static String showInputDialog(Component  
parent, String message, String title, int  
messageType)1 of class JOptionPane of package javax.swing.
```

## Question

What is the return type of the method `showInputDialog`?

## Answer

String.

## Question

How many parameters does the method `showInputDialog` have?

---

<sup>1</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

`public static String showInputDialog(Component parent, String message, String title, int messageType)`<sup>1</sup> of class `JOptionPane` of package `javax.swing`.

## Question

What is the return type of the method `showInputDialog`?

## Answer

`String`.

## Question

How many parameters does the method `showInputDialog` have?

## Answer

Four.

<sup>1</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

```
public static String showInputDialog(Component  
parent, String message, String title, int  
messageType) of class JOptionPane of package javax.swing.
```

## Question

What is the signature of the method `showInputDialog`?

---

<sup>2</sup>We will come back to `null` later in the course.

# Static methods

```
public static String showInputDialog(Component  
parent, String message, String title, int  
messageType) of class JOptionPane of package javax.swing.
```

## Question

What is the signature of the method `showInputDialog`?

## Answer

```
showInputDialog(Component, String, String, int)
```

---

<sup>2</sup>We will come back to `null` later in the course.

```
public static String showInputDialog(Component  
parent, String message, String title, int  
messageType) of class JOptionPane of package javax.swing.
```

## Question

What is the signature of the method `showInputDialog`?

## Answer

```
showInputDialog(Component, String, String, int)
```

In our case, we do not need a parent component (whatever that may be), and therefore we use the default value `null` as the first argument.<sup>2</sup>

---

<sup>2</sup>We will come back to `null` later in the course.

# Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

## Question

What is the return type of the method `parseDouble`?

# Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

## Question

What is the return type of the method `parseDouble`?

## Answer

`double`.



# Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

## Question

What is the return type of the method `parseDouble`?

## Answer

`double`.

## Question

How many parameters does the method `parseDouble` have?

# Static methods

`public static double parseDouble(String s)` of class `Double` of package `java.lang`.

## Question

What is the return type of the method `parseDouble`?

## Answer

`double`.

## Question

How many parameters does the method `parseDouble` have?

## Answer

One, of type `String`.

# Static methods

```
public static void showMessageDialog(Component  
parent, String message, String title, int  
messageType)3 of class JOptionPane of package javax.swing.
```

## Question

What is the return type of the method `showMessageDialog`?

---

<sup>3</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

```
public static void showMessageDialog(Component  
parent, String message, String title, int  
messageType)3 of class JOptionPane of package javax.swing.
```

## Question

What is the return type of the method `showMessageDialog`?

## Answer

None.

---

<sup>3</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)`<sup>3</sup> of class `JOptionPane` of package `javax.swing`.

## Question

What is the return type of the method `showMessageDialog`?

## Answer

None.

## Question

How many parameters does the method `showMessageDialog` have?

---

<sup>3</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)`<sup>3</sup> of class `JOptionPane` of package `javax.swing`.

## Question

What is the return type of the method `showMessageDialog`?

## Answer

None.

## Question

How many parameters does the method `showMessageDialog` have?

## Answer

Four.

<sup>3</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

```
public static void showMessageDialog(Component  
parent, String message, String title, int  
messageType) of class JOptionPane of package javax.swing.
```

## Question

What is the signature of the method `showMessageDialog`?

# Static methods

```
public static void showMessageDialog(Component  
parent, String message, String title, int  
messageType) of class JOptionPane of package javax.swing.
```

## Question

What is the signature of the method `showMessageDialog`?

## Answer

```
showMessageDialog(Component, String, String, int)
```



# Static methods

`public static void showMessageDialog(Component parent, String message, String title, int messageType)` of class `JOptionPane` of package `javax.swing`.

## Question

What is the signature of the method `showMessageDialog`?

## Answer

`showMessageDialog(Component, String, String, int)`

In our case, we do not need a parent component (whatever that may be), and therefore we use the default value `null` as the first argument.

# Static methods

`public static String format(String format, double value)`<sup>4</sup> of class `String` of package `java.lang`.

## Question

What is the return type of the method `format`?

---

<sup>4</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

`public static String format(String format, double value)`<sup>4</sup> of class `String` of package `java.lang`.

## Question

What is the return type of the method `format`?

## Answer

`String`.

---

<sup>4</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

`public static String format(String format, double value)`<sup>4</sup> of class `String` of package `java.lang`.

## Question

What is the return type of the method `format`?

## Answer

`String`.

## Question

How many parameters does the method `format` have?

---

<sup>4</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# Static methods

`public static String format(String format, double value)`<sup>4</sup> of class `String` of package `java.lang`.

## Question

What is the return type of the method `format`?

## Answer

`String`.

## Question

How many parameters does the method `format` have?

## Answer

Two.<sup>a</sup>

---

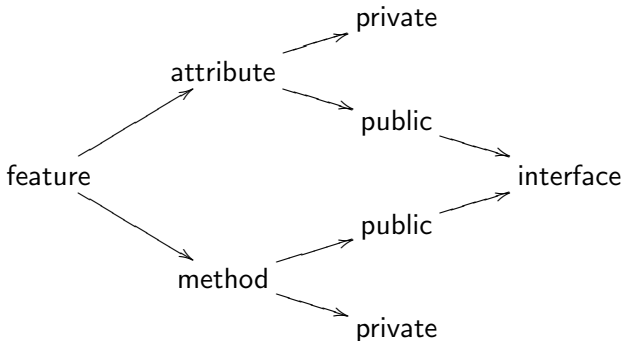
<sup>a</sup>As we will see later, the answer is “at least one.”

<sup>4</sup>The signature in the API is slightly different. We will come back to this when we cover Chapter 9 of the textbook.

# The price of gold

Write an app that pops up a dialog box with the title “The Price of Gold” and the message “Enter the amount of gold in kilos” and, after the user has entered the amount  $k$  and clicks the OK button, pops up another dialog box with the title “The Price of  $k$  kilos of Gold” and 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.”

# Some terminology



public attribute = field<sup>5</sup>

---

<sup>5</sup>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`.

---

<sup>6</sup>The textbook calls these parameters as well. On a test, you may call them either arguments or parameters.



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**,<sup>6</sup> namely `priceInUSD`, `Currency.USD` and `Currency.CAD`.

---

<sup>6</sup>The textbook calls these parameters as well. On a test, you may call them either arguments or parameters.

## Question

How do you print the string "It is Wednesday!" on the screen?

## Question

How do you print the string "It is Wednesday!" on the screen?

## Answer

```
System.out.println("It is Wednesday!");
```

## Question

- System is a

## Question

How do you print the string "It is Wednesday!" on the screen?

## Answer

```
System.out.println("It is Wednesday!");
```

## Question

- System is a **class**.
- out is an

## Question

How do you print the string "It is Wednesday!" on the screen?

## Answer

```
System.out.println("It is Wednesday!");
```

## Question

- System is a **class**.
- out is an **attribute**.
- println is a

## Question

How do you print the string "It is Wednesday!" on the screen?

## Answer

```
System.out.println("It is Wednesday!");
```

## Question

- System is a **class**.
- out is an **attribute**.
- println is a **method**.

## Question

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

## Question

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

## Answer

Study the API of the `System` class.



## 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.println("It is Wednesday!");
```

## Question

What is the signature of the `println` method in

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

## Question

What is the signature of the `println` method in  
`output.println("It is Wednesday!");`

## Answer

`println(String).`

## Question

What is the signature of the `println` method in

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

## Question

What is the signature of the `println` method in  
`output.println(123);`

## Answer

`println(int).`

## Question

What is the signature of the `println` method in

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

## Question

What is the signature of the `println` method in

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

## Answer

```
println(boolean).
```



## Question

What is the signature of the `println` method in

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

## Question

What is the signature of the `println` method in  
`output.println('\u226E');`

## Answer

`println(char).`

## Question

What is the signature of the `println` method in

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

## Question

What is the signature of the `println` method in  
`output.println()`;

## Answer

`println()`.

# Keyboard input

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

Next week we will discuss what new Scanner does.

## Question

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

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

## Question

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

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

## Answer

```
int.
```

Of course, the result should be saved in a variable.

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

## Question

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

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



## Question

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

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

## Answer

String.

Of course, the result should be saved in a variable.

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

## Question

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

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

## Question

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

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

## Answer

String.

Of course, the result should be saved in a variable.

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

## Problem

Write an app that

- prompts the user `Enter an integer:`,
- reads the integer,
- prints `You entered` followed by the integer.

```
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;

    }
}
```

**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**.<sup>7</sup>

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

---

<sup>7</sup>or path or Path or pathH, etc.

**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.cse5910`, which is stored in the jar file

[www.eecs.yorku.ca/course\\_archive/2014-15/F/5910/jar/5910.jar](http://www.eecs.yorku.ca/course_archive/2014-15/F/5910/jar/5910.jar)

save the jar file `5910.jar` and ensure that it is part of the **CLASSPATH**.

To see this list of directories and jar files, 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.

- Study Chapter 2 of the textbook.