

The price of gold



source: nowiknow.com

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 user enters a negative amount, the app crashes with the message “The amount of gold cannot be negative.”

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com
- But the price is given per Troy ounce, not per kilo. How do we address this?

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com
- But the price is given per Troy ounce, not per kilo. How do we address this?
- Using a search engine, find what a Troy ounce is.

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com
- But the price is given per Troy ounce, not per kilo. How do we address this?
- Using a search engine, find what a Troy ounce is.
- 1 Troy ounce = 31.1034768 grams

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com
- But the price is given per Troy ounce, not per kilo. How do we address this?
- Using a search engine, find what a Troy ounce is.
- 1 Troy ounce = 31.1034768 grams
- But the price is given in US dollars, not in Canadian dollars. How do we address this?

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com
- But the price is given per Troy ounce, not per kilo. How do we address this?
- Using a search engine, find what a Troy ounce is.
- 1 Troy ounce = 31.1034768 grams
- But the price is given in US dollars, not in Canadian dollars. How do we address this?
- Using a search engine, find a website that contains the current exchange rate.

Determine the price of k kilos of gold in Canadian dollars

Forget about writing an app for now. How would *you* solve this problem for $k = 0.5$?

- Using a search engine, find a website that contains the current gold price.
- www.goldpriceoz.com
- But the price is given per Troy ounce, not per kilo. How do we address this?
- Using a search engine, find what a Troy ounce is.
- 1 Troy ounce = 31.1034768 grams
- But the price is given in US dollars, not in Canadian dollars. How do we address this?
- Using a search engine, find a website that contains the current exchange rate.
- www.gocurrency.com

In our solution we use **delegation**. Instead of solving each part of the puzzle ourselves, we ask “someone else” to do it for us.

For example, we delegate to `www.goldpriceoz.com` for the current price of gold.

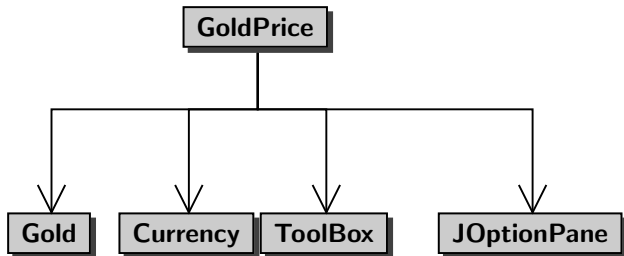
In our solution we use **delegation**. Instead of solving each part of the puzzle ourselves, we ask “someone else” to do it for us.

For example, we delegate to `www.goldpriceoz.com` for the current price of gold.

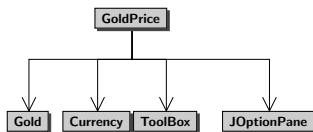
Could we do it ourselves?

Yes, we could travel to London where the price of gold is determined daily at 10.30 am and 3.00 pm, but delegation seems a little easier.

Also when writing an app, we try to delegate.



Some terminology

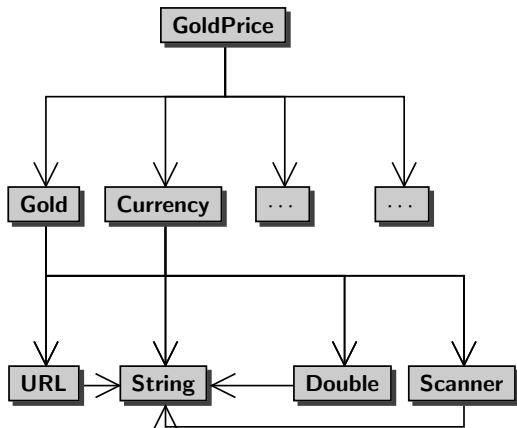


- **main class** or **app**
GoldPrice
- **helper classes** or **components**
Gold, Currency, ToolBox, and JOptionPane

The main class only contains a main method.

- **client**: developer of main class
- **implementer**: developer of components

Implementers delegate too



- 1 To solve a problem, decide what type of components are needed.
- 2 Find the appropriate components.
- 3 Delegate to the components.

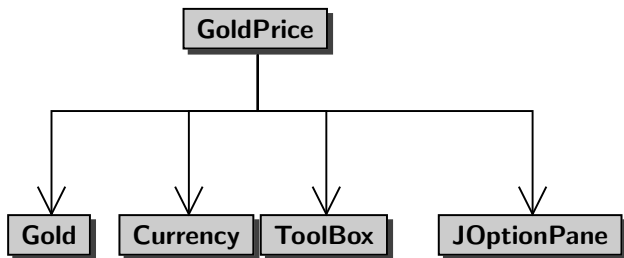
Bugs can be expensive

Flight 501, which took place on June 4, 1996, was the first, and unsuccessful, test flight of the European Ariane 5 expendable launch system. Due to an **integer overflow**, the rocket veered off its flight path 37 seconds after launch and was destroyed by its automated self-destruct system. It is one of the most infamous computer bugs in history costing roughly \$ 370,000,000.



source: spaceflightnow.com

Who is to blame for the bug?



- The user of the app (**user**)?
- The developer of the app (**client**)?
- The developer of one of the components (**implementer**)?

Interface of a component

An interface is a **contract** between the client of the component and the implementer of the component.

For each operation, it specifies

- **parameters**: “the type of data to be provided by the client to the component”
- **precondition**: “a property to be satisfied by the data provided by the client to the component”
- **postcondition**: “a property to be satisfied by the data returned by the component to the client”

The **precondition** is the **client**'s responsibility and the **postcondition** is the **implementer**'s responsibility.

Interface of `www.bankofcanada.ca/rates/exchange/daily-converter/`

parameters

amount : integer

precondition

amount $< 10^{21}$

postcondition

returns the amount converted from Canadian to US dollars

Question

Assume that the client provides -1 to the component and the component crashes. Who is to blame?

Interface of `www.bankofcanada.ca/rates/exchange/daily-converter/`

parameters

amount : integer

precondition

amount $< 10^{21}$

postcondition

returns the amount converted from Canadian to US dollars

Question

Assume that the client provides -1 to the component and the component crashes. Who is to blame?

Answer

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

Interface of `www.bankofcanada.ca/rates/exchange/daily-converter/`

parameters

amount : integer

precondition

amount $< 10^{21}$

postcondition

returns the amount converted from Canadian to US dollars

Question

Assume that the client provides 10^{21} to the component and the component crashes. Who is to blame?

Interface of `www.bankofcanada.ca/rates/exchange/daily-converter/`

parameters

amount : integer

precondition

amount $< 10^{21}$

postcondition

returns the amount converted from Canadian to US dollars

Question

Assume that the client provides 10^{21} to the component and the component crashes. Who is to blame?

Answer

The client, since the client did not provide an integer that satisfies the precondition.

Advantages of interfaces

- **Accountability**: if something goes wrong, then the interface can be used to determine who is to blame.
- **Abstraction**: the interface abstracts from many implementations details (an interface of a component is usually much simpler than the code of the component).
The interface specifies **what** the component does, **not how** it does it.
- **Substitutibility**: the implementer can change the code of the component as long as it still conforms to the interface, without affecting the client in any way.

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 user enters a negative amount, the app crashes with the message “The amount of gold cannot be negative.”

Let us start with a simplified version.

Problem

Write an app that prints the price of one Troy ounce of gold in US dollars.

Let's go component shopping!

The interface of a Java class is described by its **Application Programming Interface (API)**. Many of these APIs can be found on the Internet.

- Java Standard Library (JSL)
`docs.oracle.com/javase/7/docs/api`
- TYPE package
`www.eecs.yorku.ca/teaching/docs/type/api`
- franck.cse1020
`www.eecs.yorku.ca/course_archive/2013-14/F/1020/api/franck.cse1020.api`
- and many many more.

Convention

If the precondition is “true” (that is, it holds vacuously) then it is left out.

All classes in the JSL contain no preconditions.

Convention

If the postcondition is “returns what is specified by **Returns:**” and “crashes as specified by **Throws:**” then it is left out.

All classes in the JSL contain no postconditions.

```
public static type methodName(type1 parameterName1,  
..., typen parameterNamen)
```

- All methods we will use in our apps are **public**.
- All methods we will use today are **static**. In the near future, we will discuss methods that are not static.
- **type** is the type of the value that is returned by the method.
- **methodName** is the name of the method.
- **type_i** is the type of the parameter named **parameterName_i**.

```
public static type methodName(type1 parameterName1,  
..., typen parameterNamen)
```

- `methodName(type1, ..., typen)` is the **signature** of the method.
- `type` is the **return type** of the method.

```
public static double price()
```

Question

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

```
public static double price()
```

Question

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

Answer

`double`.


```
public static double price()
```

Question

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

Answer

`double`.

Question

How many parameters does the method `price` have?

Static methods

```
public static double price()
```

Question

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

Answer

`double`.

Question

How many parameters does the method `price` have?

Answer

Zero.

```
public static double price()
```

Question

What is the signature the method `price`?

```
public static double price()
```

Question

What is the signature the method `price`?

Answer

```
price().
```

```
public static void methodName(type1 parameterName1,  
..., typen parameterNamen)
```

- All methods we will use in our apps are **public**.
- All methods we will use today are **static**. In the near future, we will discuss methods that are not static.
- **The method does not return anything.**
- **methodName** is the name of the method.
- **type_i** is the type of the parameter named **parameterName_i**.

Invoking a static method

Consider the method `public static type methodName(type1 parameterName1, ..., typen parameterNamen)` in the class `ClassName`.

This method is invoked as

`ClassName.methodName(argument1, ..., argumentn)`

where the type of `argumenti` is (compatible with) `typei`.

Invoking a static method

Question

How do you invoke the method `price` of the class `Gold`?

Invoking a static method

Question

How do you invoke the method `price` of the class `Gold`?

Answer

```
Gold.price().
```


Invoking a static method

Question

How do you invoke the method `price` of the class `Gold`?

Answer

```
Gold.price().
```

Question

Does the method `price` return anything?

Invoking a static method

Question

How do you invoke the method `price` of the class `Gold`?

Answer

```
Gold.price().
```

Question

Does the method `price` return anything?

Answer

Yes.

Invoking a static method

Question

How do you invoke the method `price` of the class `Gold`?

Answer

```
Gold.price().
```

Question

Does the method `price` return anything?

Answer

Yes.

Question

Should we store the result in a variable?

Invoking a static method

Question

How do you invoke the method `price` of the class `Gold`?

Answer

```
Gold.price().
```

Question

Does the method `price` return anything?

Answer

Yes.

Question

Should we store the result in a variable?

Answer

Yes.

The price of gold

Problem

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

```
public static type attributeName
```

- All attributes we will use in our apps are `public`.
- All attributes we will use in our apps are `static`.
- `type` is the type of the attribute.

```
public static final type attributeName
```

- All attributes we will use in our apps are **public**.
- All attributes we will use in our apps are **static**.
- **The attribute is a constant.**
- **type** is the type of the attribute.

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Answer

`double`.

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Answer

`double`.

Question

Is the attribute `GRAMS_PER_TROY_OUNCE` a constant?

Static attributes

```
public static final double GRAMS_PER_TROY_OUNCE
```

Question

What is the type of the attribute `GRAMS_PER_TROY_OUNCE`?

Answer

double.

Question

Is the attribute `GRAMS_PER_TROY_OUNCE` a constant?

Answer

Yes.

Using a static attribute

Consider the attribute `public static type attributeName` in the class `className`.

The attribute is used as `className.attributeName`.

Question

How do you use the attribute `GRAMS_PER_TROY_OUNCE` of the class `Gold`?

Using a static attribute

Question

How do you use the attribute `GRAMS_PER_TROY_OUNCE` of the class `Gold`?

Answer

```
Gold.GRAMS_PER_TROY_OUNCE
```

Code convention for names of constants

- Use uppercase characters.
- If the name is made up of more than one word, separate the words by an underscore.

Memory model

0		
1		
⋮		
8	Gold.main	
	1000	GRAMS_PER_KILO
	32.150746	ouncePerKilo
	42175.349	price
⋮		
112	Gold	
	31.103476	GRAMS_PER_TROY_OUNCE
⋮		

Problem

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

```
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.

```
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.

The price of gold

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`?

¹The 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.

¹The 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?

¹The 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?

Answer

Four.

¹The 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`?

²We 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)
```

²We 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.²

²We 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`?

³The 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.

³The 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.

Question

How many parameters does the method `showMessageDialog` have?

³The 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.

Question

How many parameters does the method `showMessageDialog` have?

Answer

Four.

³The 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`?

```
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)
```



```
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)`⁴ of class `String` of package `java.lang`.

Question

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

⁴The 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)`⁴ of class `String` of package `java.lang`.

Question

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

Answer

`String`.

⁴The 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)`⁴ 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?

⁴The 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)`⁴ 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.^a

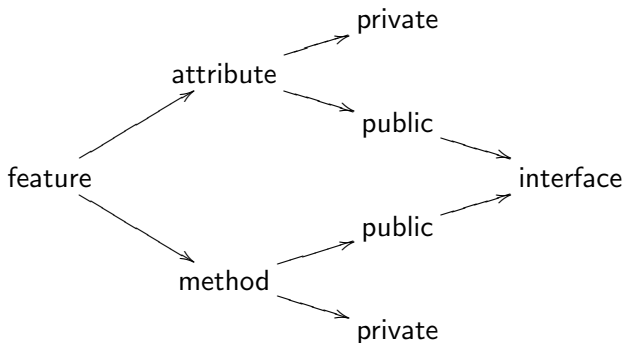
^aAs we will see later, the answer is “at least one.”

⁴The 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 = field⁵

⁵Not 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`.

⁶The 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**,⁶ namely `priceInUSD`, `Currency.USD` and `Currency.CAD`.

⁶The 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`.

```
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()`.

```
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**.⁷

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

⁷or 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.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.

- Study Chapter 2 of the textbook.
- Complete Check02A from the textbook before January 25.