Age of Java

Write a Java app that prints Java's age.

```
public class {
    public static void main(String[] args) {
        ...
    }
}
```

Body of main method

```
int birthYear = 1995;
int currentYear = 2014;
int year = currentYear - birthYear;

int birthDay = 143;
int currentDay = 13;
int day = currentDay - birthDay;

int daysPerYear = 365;

double age = year + day / (double) daysPerYear;

System.out.print("The age of Java is: %.2f\n", age);
```

Questions

Question

What is casting?

Questions

Question

What is casting?

Answer

A unary operator that converts a value of one type to a corresponding value of another type.

Questions

Question

What is the type of the operator $(\text{double}) \cdot$ in $(\text{double})\ \text{year} + \cdots$?

Questions

Question

What is the type of the operator $(\text{double}) \cdot$ in $(\text{double})\ \text{year} + \cdots$?

Answer

$(\text{double}) \cdot : \text{int} \longrightarrow \text{double}$

Questions

Question

Can the first cast in $(\text{double})\ \text{year} + (\text{double}) \cdots$ be left out? Explain why or why not.

Questions

Question

Can the first cast in $(\text{double})\ \text{year} + (\text{double}) \cdots$ be left out? Explain why or why not.

Answer

Yes. In that case, $(\text{double}) \cdots$ is of type double and year is of type int, and, hence, the compiler will add the cast (double) automatically.

Questions

Question

Can one of the casts in (double) day / (double) daysPerYear be left out? Explain why or why not.

Questions

Question

Can one of the casts in $(\text{double})\ \text{day}\ /\ (\text{double})\ \text{daysPerYear}$ be left out? Explain why or why not.

Answer

Yes. In that case, one of the expressions is of type double and the other is of type int, and, hence, the compiler will add the cast (double) automatically.

Questions

Question

Can both casts in (double) day / (double) daysPerYear be left out? Explain why or why not.

Questions

Question

Can both casts in (double) day / (double) daysPerYear be left out? Explain why or why not.

Answer

No. In that case, both expressions are of type int, and, hence, the division operator / of type int $\times$ int $\rightarrow$ int is used, which does not give the desired result.

Another problem

Write an app that prints the age of Java as a real number preceded by

```
The age of Java is
```

which not only gives the correct result today, but also tomorrow, the day after tomorrow, etc.

Importing packages

```
import franck.cse1020.Today;
```

- franck is a package
- cse1020 is a subpackage
- Today is a class

Another problem

Write an app that prints the age of Java as a real number preceded by

```
The age of Java is
```

which not only gives the correct result today, but also tomorrow, the day after tomorrow, etc, even if it is a leap year.

Remainder operator

The operator

$$\cdot \% \cdot : (\text{int} \times \text{int}) \rightarrow \text{int}$$

yields the remainder of the division.

For example, the expression $2014 \ \% \ 4$ evaluates to 2.

Property

For all values a and b of type int,

$$(\text{a} \ / \ \text{b}) * \text{b} + (\text{a} \ \% \ \text{b}) = \text{a}$$

Data types: example

- name: boolean
- values: true, false
- operations:

$$\cdot \&\& \cdot : (\text{boolean} \times \text{boolean}) \rightarrow \text{boolean}$$
$$\cdot || \cdot : (\text{boolean} \times \text{boolean}) \rightarrow \text{boolean}$$
$$! \cdot : \text{boolean} \rightarrow \text{boolean}$$
$$\ldots$$

Some binary operations

$$
\begin{aligned}
&\cdot == \cdot : (\text{int} \times \text{int}) \to \text{boolean} \\
&\cdot < \cdot : (\text{int} \times \text{int}) \to \text{boolean} \\
&\cdot <= \cdot : (\text{int} \times \text{int}) \to \text{boolean} \\
&\ldots \\
&\cdot == \cdot : (\text{double} \times \text{double}) \to \text{boolean} \\
&\cdot < \cdot : (\text{double} \times \text{double}) \to \text{boolean} \\
&\cdot <= \cdot : (\text{double} \times \text{double}) \to \text{boolean} \\
&\ldots
\end{aligned}
$$

The expression $5 == 6$ evaluates to false and the expression $5 <= 6$ evaluates to true.

A ternary operation

The operation

$$\cdot ? \cdot : \cdot$$

of type

$$(\text{boolean} \times \text{int} \times \text{int}) \rightarrow \text{int}$$

is ternary, since it takes three arguments.

The expression $(5 == 6)\ ?\ 0 : 1$ evaluates to 1 and the expression $(5 <= 6)\ ?\ 0 : 1$ evaluates to 0.

Another problem

Write an app that prints the age of Java as a fraction preceded by

```
The age of Java is
```

which not only gives the correct result today, but also tomorrow, the day after tomorrow, etc, even if it is a leap year.

Sample question for a test

This question is based on material that has not been covered in the lectures, but can be found in the textbook.

Question

We can denote the steps that take place when evaluating the expression $(3 + 4) - 2$ as follows:

$$(3 + 4) - 2 \longrightarrow 7 - 2 \longrightarrow 5$$

Give the steps that take place when evaluating the expression

$$5 + (4 - 3)/5 - 2 * 3\%4$$

Test1

- When: Friday January 17, during the lab (14:30–16:00)
- Where: Lassonde building, labs 1006, 1004, 1002
- Material: Chapter 1 of the textbook
- What: One programming question similar to Check01C and several conceptial questions

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 users 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`

Delegation

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.

Delegation

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

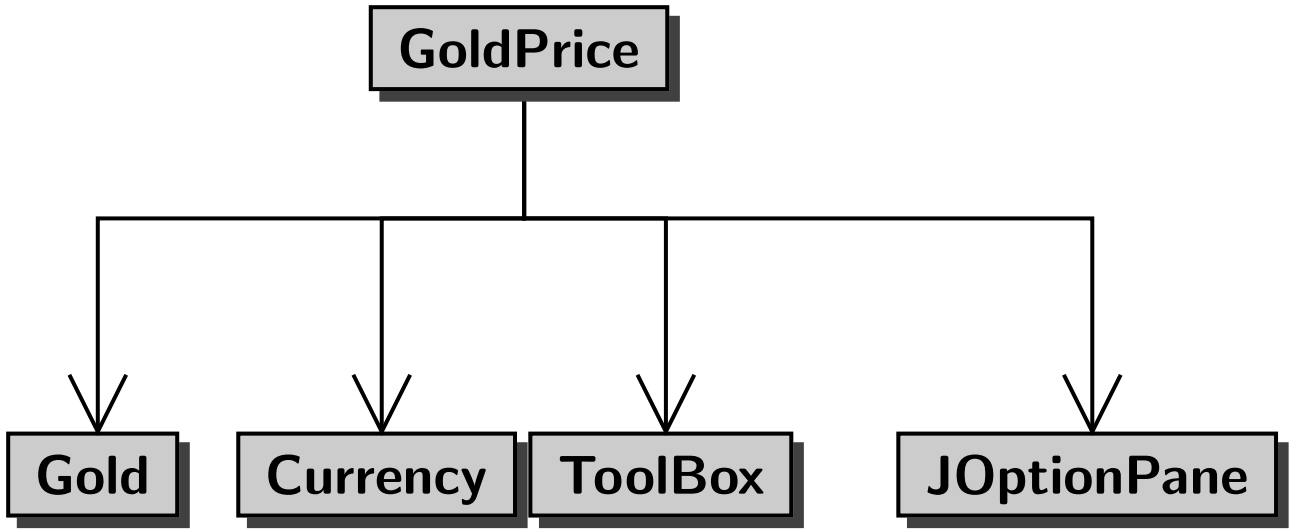
For example, we delegate to `www.goldpriceoz.com` for the current price of gold.

Could we do it ourselves?

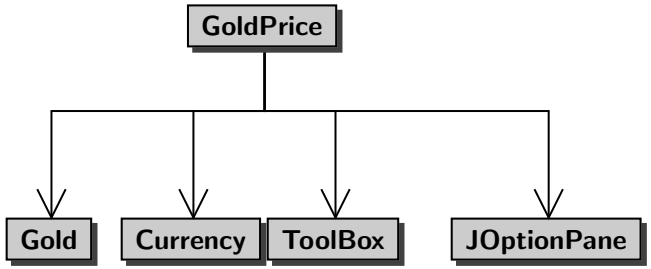
Yes, we could travel to London where to price of gold is determined daily at 10.30 am and 3.00 pm, but delegation seems a little easier.

Delegation

Also when writing an app, we try to delegate.



Some terminology

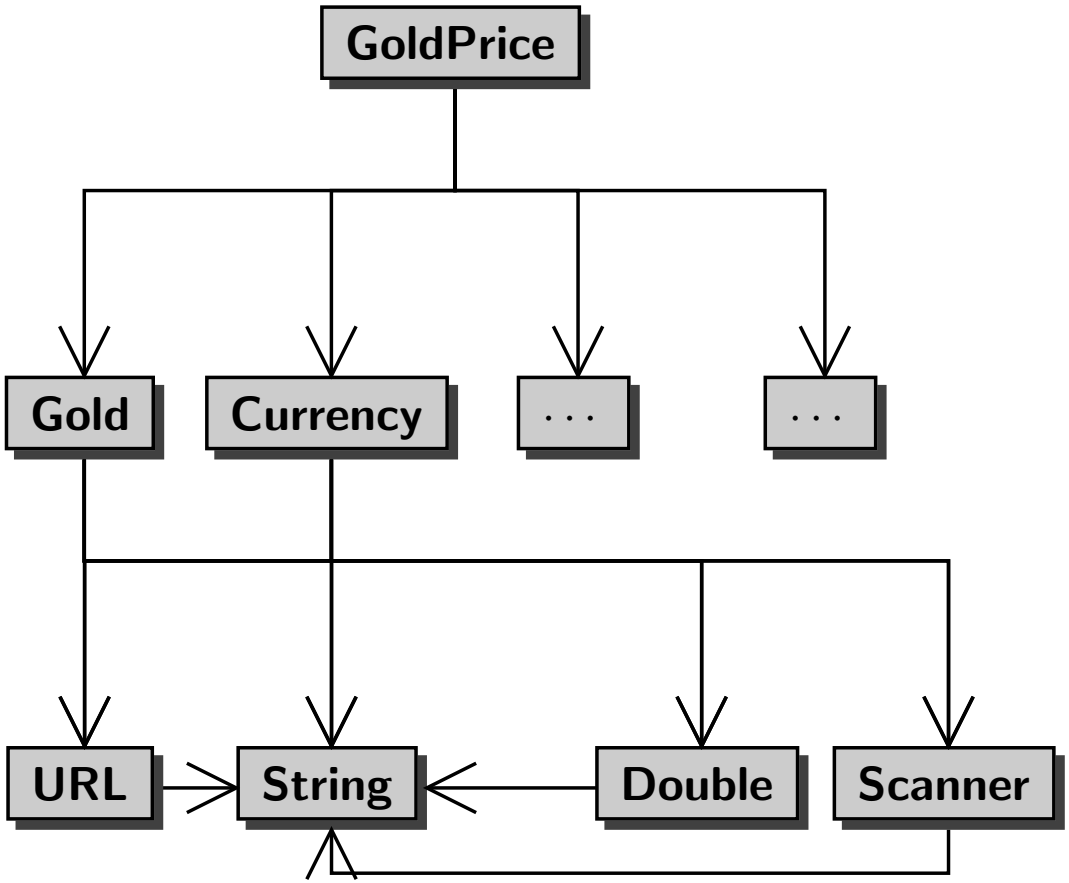


- main class or app
  GoldPrice
- helper classes or components
  Gold, Currency, ToolBox, and JOptionPane

The main class only contains a main method.

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

Implementers delegate too

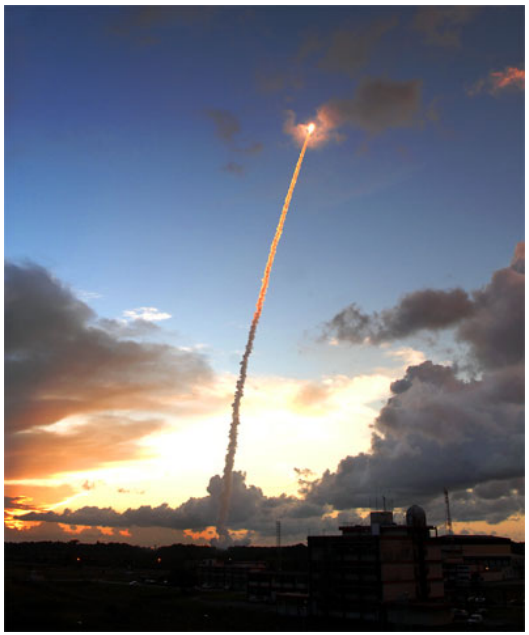


Client approach

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

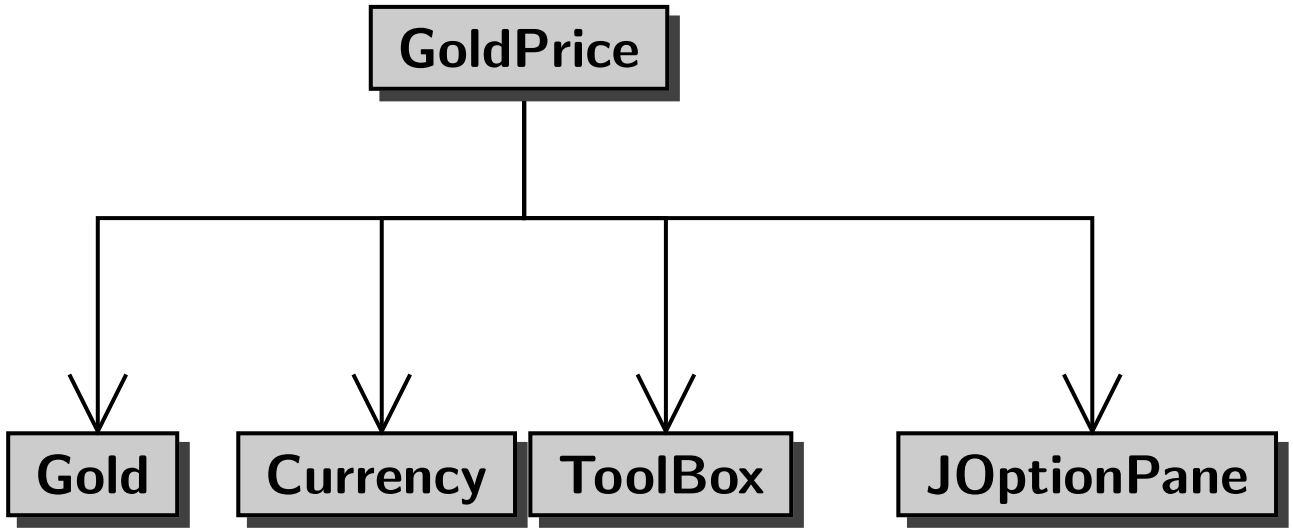
Bugs can be expensive

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 users enters a negative amount, the app crashes with the message "The amount of gold cannot be negative."

The Price of Gold

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.

Pre- and postconditions in JSL

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.

Static methods

`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.
- $\texttt{type}_i$ is the type of the parameter named $\texttt{parameterName}_i$.

Terminology

`public static type methodName(type`$_1$ `parameterName`$_1$`, ..., type`$_n$ `parameterName`$_n$`)`

- `methodName(type`$_1$`, ..., type`$_n$`)` is the signature of the method.
- `type` is the return type of the method.

Static methods

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

Question

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

Static methods

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

Question

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

Answer

`double`.

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?

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.

Static methods

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

Question

What is the signature the method `price`?

Static methods

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

Question

What is the signature the method `price`?

Answer

`price()`.

Static methods

`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.
- `typei` is the type of the parameter named `parameterNamei`.

Invoking a static method

Consider the method `public static type methodName(type`$_1$ `parameterName`$_1$`, ..., type`$_n$ `parameterName`$_n$`)` in the class `ClassName`.

This method is invoked as
`ClassName.methodName(argument`$_1$`, ..., argument`$_n$`)`
where the type of `argument`$_i$ is (compatible with) `type`$_i$.

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.

To do

- Study Section 2.1.1, 2.1.4, 2.2.1, 2.2.2 and 2.3.3 of the textbook.
- Activate your EECS account: `www.eecs.yorku.ca/activ8`.