Static methods

Question

How do you invoke the static method pow of the class Math to compute 2^1?

Static methods

Question

How do you invoke the static method pow of the class Math to compute 2^1?

Answer

Math.pow(2, 1)

Static methods

Question

How do you invoke the static method pow of the class Math to compute 2^1?

Answer

Math.pow(2, 1)

Question

What should you do with the result?

Static methods

Question

How do you invoke the static method pow of the class Math to compute 2^1?

Answer

Math.pow(2, 1)

Question

What should you do with the result?

Answer

Store it in a variable.

Static attributes

Question

How do you use the static attribute PI of the class Math?

Static attributes

Question

How do you use the static attribute PI of the class Math?

Answer

$\mathrm{Math.PI}$

Question

Draw the memory diagram for the main method with body

$$\text{double radius} = 1.0;$$
$$\text{double area} = \text{Math.PI} * \text{radius} * \text{radius};$$

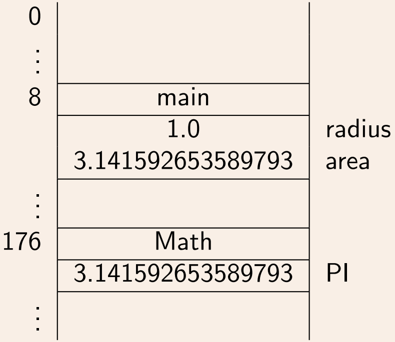
Static attributes

Question

Draw the memory diagram for the main method with body

double radius = 1.0;
double area = Math.PI * radius * radius;

Answer



Programming Paradigms

- Object-oriented programming
- Imperative programming
- Functional programming
- Logic programming
- Concurrent programming
- Event-driven programming
- Constraint programming
- ...

Object-Oriented Programming

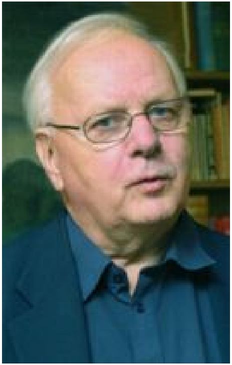
Objects as a formal concept in programming were introduced in the 1960s in programming language Simula 67. This language was created by Ole-Johan Dahl and Kristen Nygaard of the Norwegian Computing Center in Oslo.

Ole-Johan Dahl

Ole-Johan Dahl (October 12, 1931 – June 29, 2002) was a Norwegian computer scientist and is considered to be one of the fathers of object-oriented programming.

source: ifi.uio.no

Kristen Nygaard (August 27, 1926 – August 10, 2002) was a Norwegian computer scientist and is considered to be one of the fathers of object-oriented programming.



source: ifi.uio.no

Dahl and Nygaard

In 2001, Ole-Johan Dahl and Kristen Nygaard won the Turing award.

The A.M. Turing Award is given annually by the Association for Computing Machinery (ACM) to "an individual selected for contributions of a technical nature made to the computing community." The Turing Award is recognized as the "highest distinction in Computer Science" and "Nobel Prize of computing."

source: ifi.uio.no

Advantages of OOP

- easy to re-use code
- easy to extend code
- easy to maintain code
- easy to test code
- fits well with the real world
- ...

However, (some of) these advantages are debatable.

Mordechai Ben-Ari. Objects never?: well, hardly ever! *Communications of the ACM*, 53(9): 32–35, September 2010.

Round off

Question

Does the following snippet produce 1.0 as output?

```
double one = 1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0;
output.println(one);
```

Round off

Question

Does the following snippet produce 1.0 as output?

```
double one = 1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0 +
    1.0 / 7.0;
output.println(one);
```

Answer

No.

Primitive types

Question

What are the names of the five most used primitive types?

Primitive types

Question

What are the names of the five most used primitive types?

Answer

boolean, char, double, int and long.[a]

[a] The other three, less used, primitive types are byte, float and short.

None of these can represent 1.0 / 7.0 exactly.

How to represent fractions?

Question

You want to record a fraction, say $\frac{1}{7}$. What kind of data would you record?

How to represent fractions?

Question

You want to record a fraction, say $\frac{1}{7}$. What kind of data would you record?

Answer

- the numerator and
- the denominator.

How to represent fractions?

Question

You want to record a fraction, say $\frac{1}{7}$. What kind of data would you record?

Answer

- the numerator and
- the denominator.

Question

For each datum, what is a descriptive name and an appropriate type?

How to represent fractions?

Question

You want to record a fraction, say $\frac{1}{7}$. What kind of data would you record?

Answer

- the numerator and
- the denominator.

Question

For each datum, what is a descriptive name and an appropriate type?

Answer

- numerator : long
- denominator : long

How to represent fractions?

Question

How to represent $\frac{1}{7}$?

How to represent fractions?

Question

How to represent $\frac{1}{7}$?

Answer

numerator	1
denominator	7

How to represent fractions?

Question

How to represent $\frac{1}{7}$?

Answer

numerator	1
denominator	7

Question

How to represent $\frac{3}{4}$?

How to represent fractions?

Question

How to represent $\frac{1}{7}$?

Answer

numerator	1
denominator	7

Question

How to represent $\frac{3}{4}$?

Answer

numerator	3
denominator	4

How to represent fractions?

All fractions are an instance of the following pattern.

numerator
denominator

How to manipulate fractions?

If you are given an instance of the pattern

$$\frac{\text{numerator}}{\text{denominator}} \quad \boxed{}$$

what kind of questions may you want to ask about this data?

How to manipulate fractions?

If you are given an instance of the pattern

numerator	
denominator	

what kind of questions may you want to ask about this data?

- What is the numerator of this fraction?

How to manipulate fractions?

If you are given an instance of the pattern

numerator	
denominator	

what kind of questions may you want to ask about this data?

- What is the numerator of this fraction?
- What is the denominator of this fraction?

How to manipulate fractions?

If you are given an instance of the pattern

numerator	
denominator	

what kind of questions may you want to ask about this data?

- What is the numerator of this fraction?
- What is the denominator of this fraction?
- What is the sum of this fraction and another fraction?

How to manipulate fractions?

If you are given an instance of the pattern

numerator	
denominator	

what kind of questions may you want to ask about this data?

- What is the numerator of this fraction?
- What is the denominator of this fraction?
- What is the sum of this fraction and another fraction?
- What is the product of this fraction and another fraction?
- ...

Objects and classes

Question

What is an object?

Answer

"An instance of a class."

Question

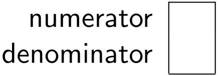
What is a class?

Answer

"A blueprint for objects."

You often find these circular definitions in textbooks and on the Internet, but they are not particularly helpful.

What is a class?



A class contains (non-static) attributes. Each attribute has a name and a type.

numerator : long
denominator : long

What is a class?

- What is the numerator of this fraction?
- What is the denominator of this fraction?
- ...

A class contains (non-static) methods. Each method has a signature and possibly a return type.

getNumerator() : long
getDenominator() : long

What is an object?

An object is an instance of a class.

An object has a state. The state of an object consists of the non-static attributes of the class and their values.

numerator	1
denominator	7

What is an object?

An object has an identity. This identity is unique. That is, two different objects have different identities.

This is an abstract notion. In more concrete terms, you may think of an object's identity as the address in memory where it is stored. Obviously, two different objects cannot be stored at the same memory address.

What is a class?

A class contains constructors. Each constructor has a signature, the name of which is the same as the name of the class.

Fraction()
Fraction(long, long)

API

The API of the Fraction class contains

- constructors and
- methods.

Question

The class Fraction has attributes numerator and denominator. Why are these attributes not present in the API?

API

The API of the Fraction class contains

- constructors and
- methods.

Question

The class Fraction has attributes numerator and denominator. Why are these attributes not present in the API?

Answer

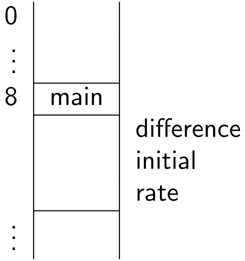
The attributes numerator and denominator are private.

How to create objects?

```
output.print("Enter difference of initial and final value: ");
long difference = input.nextLong();
output.print("Enter initial value: ");
long initial = input.nextLong();
Fraction rate = new Fraction(difference, initial);
```

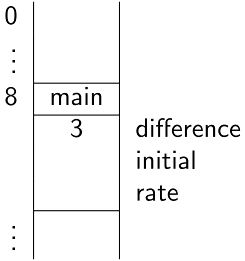
How to create objects?

```
long difference = 3;
long initial = 4;
Fraction rate = new Fraction( difference , initial );
```



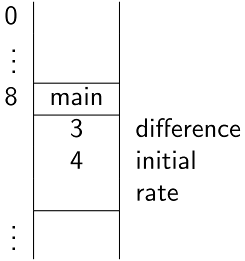
How to create objects?

```
long difference = 3;
long initial = 4;
Fraction rate = new Fraction( difference , initial );
```



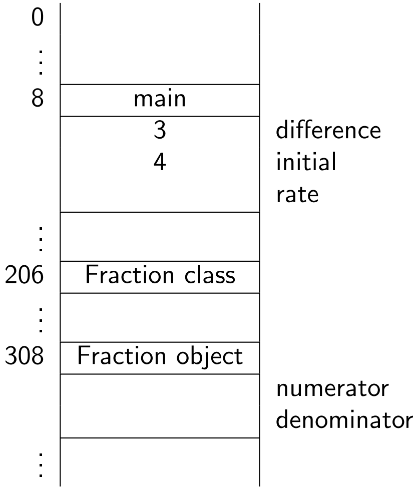
How to create objects?

```
long difference = 3;
long initial = 4;
Fraction rate = new Fraction( difference , initial );
```



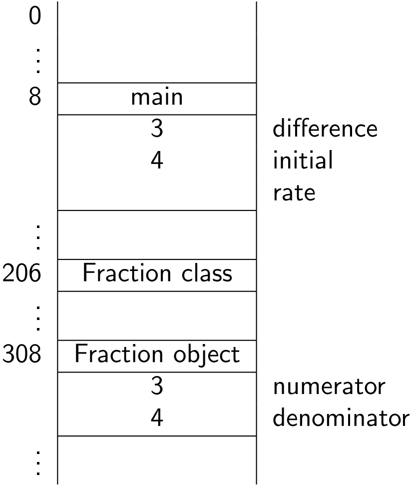
How to create objects?

```
long difference = 3;
long initial = 4;
Fraction rate = new Fraction(difference, initial);
```



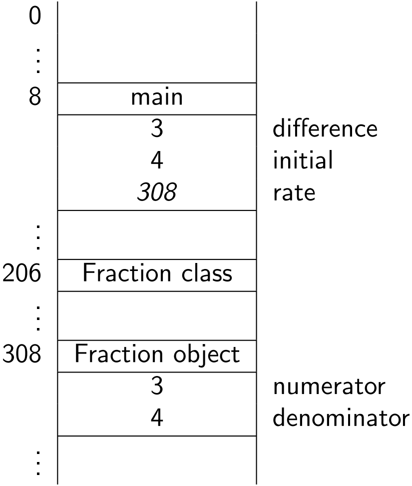
How to create objects?

```
long difference = 3;
long initial = 4;
Fraction rate = new Fraction(difference, initial);
```



How to create objects?

```
long difference = 3;
long initial = 4;
Fraction rate = new Fraction(difference, initial);
```



Object creation in memory model

- The first time we encounter a class, we allocate a block in memory for the class.
- Whenever we encounter new, we allocate a block in memory for the object.
- Whenever we encounter a constructor, we initialize the attributes by putting the values of the attributes in the block of the object.

Terminology

```
long numerator = 1;
long denominator = 7;
Fraction fraction = new Fraction(numerator, denominator);
```

- fraction is the name of a

Terminology

$$\text{long numerator} = 1;$$
$$\text{long denominator} = 7;$$
$$\text{Fraction fraction} = \text{new Fraction(numerator, denominator)};$$

- fraction is the name of a variable.

Terminology

```
long numerator = 1;
long denominator = 7;
Fraction fraction = new Fraction(numerator, denominator);
```

- fraction is the name of a variable.
- the type of the variable fraction is

Terminology

```
long numerator = 1;
long denominator = 7;
Fraction fraction = new Fraction(numerator, denominator);
```

- fraction is the name of a variable.
- the type of the variable fraction is Fraction.

Terminology

```
long numerator = 1;
long denominator = 7;
Fraction fraction = new Fraction(numerator, denominator);
```

- fraction is the name of a variable.
- the type of the variable fraction is Fraction.
- fraction is also called an object reference.

Types

We distinguish between

- primitive types: boolean, char, double, int, long, (byte, float, short) and
- reference types: classes

Compute $\frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Compute $\frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Answer

Two.[a]

[a] Although it can be done with one.

Compute $\frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Answer

Two.[a]

[a] Although it can be done with one.

Question

Once we have those two objects, which method do we use to add them?

Compute $\frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Answer

Two.[a]

[a] Although it can be done with one.

Question

Once we have those two objects, which method do we use to add them?

Answer

The add method.

Compute $\frac{1}{7} + \frac{1}{7}$

Question

How do you create $\mathrm{Fraction}$ objects named first and second which each represent $\frac{1}{7}$?

Compute $\frac{1}{7} + \frac{1}{7}$

Question

How do you create Fraction objects named first and second which each represent $\frac{1}{7}$?

Answer

$$\begin{array}{l} \text{long numerator} = 1; \\ \text{long denominator} = 7; \\ \text{Fraction first} = \text{new Fraction(numerator, denominator)}; \\ \text{Fraction second} = \text{new Fraction(numerator, denominator)}; \end{array}$$

Compute $\frac{1}{7} + \frac{1}{7}$

Question

Draw the diagram representing the memory once the execution has reached the end of the following snippet.

$$\begin{array}{l}
\text{long numerator} = 1; \\
\text{long denominator} = 7; \\
\text{Fraction first} = \text{new Fraction(numerator, denominator)}; \\
\text{Fraction second} = \text{new Fraction(numerator, denominator)};
\end{array}$$

Answer

⋮		
8	main	
	1	numerator
	7	denominator
	308	first
	408	second
⋮		
206	Fraction class	
⋮		
308	Fraction object	
	1	numerator
	7	denominator
⋮		
408	Fraction object	
	1	numerator
	7	denominator
⋮		

Invoking a non-static method

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

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

Invoking a non-static method

```
long numerator = 1;
long denominator = 7;
Fraction first = new Fraction(numerator, denominator);
Fraction second = new Fraction(numerator, denominator);
```

Question

How do you invoke `public void add(Fraction other)` to add `second` to `first`?

Invoking a non-static method

```
long numerator = 1;
long denominator = 7;
Fraction first = new Fraction(numerator, denominator);
Fraction second = new Fraction(numerator, denominator);
```

Question

How do you invoke `public void add(Fraction other)` to add `second` to `first`?

Answer

```
first.add(second)
```

Invoking a non-static method

The invocation

```
first.add(second)
```

contains two object references:

- `first` is (a reference to) the object on which the method is invoked, and
- `second` is (a reference to) the object that is provided as an argument to the method.

Invoking a non-static method

⋮		
308	Fraction object	
	1	numerator
	7	denominator
⋮		
408	Fraction object	
	1	numerator
	7	denominator
⋮		
452	add	
	308	this
	408	other
⋮		

Invoking a non-static method

Question

Does the method

`public void add(Fraction other)`

return anything?

Invoking a non-static method

Question

Does the method

`public void add(Fraction other)`

return anything?

Answer

No.

Invoking a non-static method

Question

Does the method

`public void add(Fraction other)`

return anything?

Answer

No.

Question

If it does not return anything, does it do anything?

Invoking a non-static method

Question

Does the method

```
public void add(Fraction other)
```

return anything?

Answer

No.

Question

If it does not return anything, does it do anything?

Answer

Yes, it changes the state of the object on which it is invoked.

Invoking a non-static method

⋮		
308	Fraction object	
	2	numerator
	7	denominator
⋮		
408	Fraction object	
	1	numerator
	7	denominator
⋮		

Compute $\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}$

Question

How many objects do we need?

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Answer

Two.

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Answer

Two.

Question

Once we have those two objects, which method do we use to add them?

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

Question

How many objects do we need?

Answer

Two.

Question

Once we have those two objects, which method do we use to add them?

Answer

The add method.

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

```
long numerator = 1;
long denominator = 7;
Fraction seventh = new Fraction(numerator, denominator);
Fraction sum = new Fraction();
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
```

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

Question

Is there a method we can use to print the result?

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

Question

Is there a method we can use to print the result?

Answer

Yes, `public String toString()`

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

Question

Is there a method we can use to print the result?

Answer

Yes, `public String toString()`

Question

How do we invoke this method?

Question

Is there a method we can use to print the result?

Answer

Yes, `public String toString()`

Question

How do we invoke this method?

Answer

`String result = sum.toString()`

Compute $\frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}$

```
long numerator = 1;
long denominator = 7;
Fraction seventh = new Fraction(numerator, denominator);
Fraction sum = new Fraction();
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
sum.add(seventh);
String result = sum.toString();
output.println(result);
```

Exercise

Draw the diagram representing the memory once the execution has reached the end of the snippet on the previous slide.

Solution to exercise

100	main	
	1	numerator
	7	denominator
	300	seventh
	400	sum
	600	result
200	Fraction class	
300	Fraction object	
	1	numerator
	7	denominator
400	Fraction object	
	1	numerator
	1	denominator
500	String class	
600	String object	
	"1/1"	

Although input and output are also stored in memory, we usually do not draw them.

Check whether $\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}$ is 1

To check whether $\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}+\frac{1}{7}$ is equal to 1, let us first contrast ...

Question

```
Fraction f = new Fraction();
Fraction g = new Fraction();
Fraction h = new Fraction(1, 2);
Fraction i = new Fraction(0, 2);
Fraction j = g;
Fraction k = j;
```

At the end of the execution of the above snippet, how many objects are there and how many objects references are there?

Question

```
Fraction f = new Fraction();
Fraction g = new Fraction();
Fraction h = new Fraction(1, 2);
Fraction i = new Fraction(0, 2);
Fraction j = g;
Fraction k = j;
```

At the end of the execution of the above snippet, how many objects are there and how many objects references are there?

Answer

Four objects and six object references.

Question

```
Fraction f = new Fraction();
Fraction g = new Fraction();
Fraction h = new Fraction(1, 2);
Fraction i = new Fraction(0, 2);
Fraction j = g;
Fraction k = j;
```

At the end of the execution of the above snippet, how many objects are there and how many objects references are there?

Answer

Four objects and six object references.

Exercise

Draw the diagram representing the memory once the execution has reached the end of the above snippet.

Solution to exercise

100	main	
	300	f
	400	g
	500	h
	600	i
	400	j
	400	k
200	Fraction class	
300	Fraction object	
	0	numerator
	1	denominator
400	Fraction object	
	0	numerator
	1	denominator
500	Fraction object	
	1	numerator
	2	denominator
600	Fraction object	
	0	numerator
	2	denominator