Bethune College presents ...

- Peer Assisted Study Sessions (PASS)
- Class Representatives

See Moodle for more details.

York Programming Contests

Will you be the next York Programming Champion?

York Programming Contests

Will you be the next York Programming Champion?

To qualify for the final, come and compete on

- Thursday January 15, 16:00-18:00
- Tuesday January 20, 10:00-12:00
- Monday January 26, 14:00-16:00
- Tuesday February 3, 10:00-12:00
- Friday February 27, 18:00-20:00
- Thursday March 5, 16:00-18:00

The final will take place on Friday March 13, 17:00-20:00. All contests take place in Lab 1004 of the Lassonde Building.

York Programming Contests

Why participate?

- For fun.
- To develop your programming skills (in each contest, there will be one question related to material covered in each of the following courses: 1020, 1030, 2011 and 3101).
- To meet your fellow students.
- To develop your problem-solving skills.
- To prepare and qualify for the ACM Programming Contest.

York Programming Contests

More information can be found at
https://wiki.eecs.yorku.ca/project/ACM/

Join our facebook group
http://www.facebook.com/group.php?gid=33839119950

Last lecture

Problem

Implement this API of the `Converter` class.

Let's have a look at the API

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

Question

How is a static method invoked?

Let's have a look at the API

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

Question

How is a static method invoked?

Answer

On the class.

Let's have a look at the API

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

Question

Does the `convert` method return anything?

Let's have a look at the API

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

Question

Does the `convert` method return anything?

Answer

Yes, a value of type `double`.

Let's have a look at the API

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

Question

How many parameters does the `convert` method have?

Let's have a look at the API

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

Question

How many parameters does the `convert` method have?

Answer

Three, of types `double`, `int` and `int`.

Let's have a look at the API

```
Precondition:
    amount >= 0,
    from == Converter.CENTIMETER || from == ...
    to == Converter.CENTIMETER || to == ...
```

Question

Whose responsibility is the precondition, the client or the implementer?

Let's have a look at the API

```
Precondition:
    amount >= 0,
    from == Converter.CENTIMETER || from == ...
    to == Converter.CENTIMETER || to == ...
```

Question

Whose responsibility is the precondition, the client or the implementer?

Answer

The client.

Let's have a look at the API

```
Precondition:
    amount >= 0,
    from == Converter.CENTIMETER || from == ...
    to == Converter.CENTIMETER || to == ...
```

Question

If the client provides arguments that do not satisfy the precondition (for example, a negative amount), as implementer, what should we do?

Let's have a look at the API

```
Precondition:
    amount >= 0,
    from == Converter.CENTIMETER || from == ...
    to == Converter.CENTIMETER || to == ...
```

Question

If the client provides arguments that do not satisfy the precondition (for example, a negative amount), as implementer, what should we do?

Answer

Anything we like!

Let's have a look at the API

```
Returns:
    the given amount converted from the given unit to
    the other given unit.
```

Question

Whose responsibility is the postcondition, the client or the implementer?

Let's have a look at the API

```
Returns:
    the given amount converted from the given unit to
    the other given unit.
```

Question

Whose responsibility is the postcondition, the client or the implementer?

Answer

The implementer.

Question

How do we compute the converted amount?

Design

Question

How do we compute the converted amount?

Answer

Figure this out on a piece of paper.

Implementation

Now start eclipse and implement your algorithm.

Implementation

Now start eclipse and implement your algorithm.
We need one new ingredient: the `return` statement.

The return statement

```
public static T m()
{
    ...
    return e;
}
```

The type of the expression e has to be compatible with the type T.

The return statement

Question

```
public static int m()
{
    ...
    return 0;
}
```

Why does the above satisfy the compatibility constraint?

The return statement

Question

```
public static int m()
{
    ...
    return 0;
}
```

Why does the above satisfy the compatibility constraint?

Answer

Because the value 0 is of type `int`.

The return statement

Question

```
public static double m()
{
    ...
    return 0;
}
```

Why does the above satisfy the compatibility constraint?

The return statement

Question

```
public static double m()
{
    ...
    return 0;
}
```

Why does the above satisfy the compatibility constraint?

Answer

Because the value 0 can be promoted to a double.

The return statement

Question

```
public static Object m()
{
    ...
    return "zero";
}
```

Why does the above satisfy the compatibility constraint?

The return statement

Question

```
public static Object m()
{
    ...
    return "zero";
}
```

Why does the above satisfy the compatibility constraint?

Answer

Because is a `String` is-an `Object`.

Question

Do we have to create the `factor` map every time we invoke the `convert` method?

Implementation

Question

Do we have to create the `factor` map every time we invoke the `convert` method?

Answer

Ideally, we should only have to create it once.

Implementation

Question

If we only create it once, how do you store the `factor` map?

Implementation

Question

If we only create it once, how do you store the `factor` map?

Answer

As an attribute (its scope is the whole class and hence include the `convert` method).

Implementation

Question

Is the attribute static or not?

Implementation

Question

Is the attribute static or not?

Answer

Static, since the map is associated with the class.

Question

Is the attribute public or private?

Implementation

Question

Is the attribute public or private?

Answer

Private, since it is not part of the API.

Question

How do we initialize this attribute?

Implementation

Question

How do we initialize this attribute?

Answer

In a static initializer (these blocks are executed when a class in loaded into memory).

Documentation

To document parameters, preconditions and return, we use the tags `@param`, `@pre.` (user defined) and `@return`

Software development

Question

We have done

- analysis
- design
- implementation

What is next?

Software development

Question

We have done

- analysis
- design
- implementation

What is next?

Question

Testing.

Testing

Question

For testing, we design a ..., which consists of multiple ....

Testing

Question

For testing, we design a ..., which consists of multiple ....

Answer

For testing, we design a test suite/test vector, which consists of multiple test cases.

Question

To test the `convert` method, what does a test case consist of?

Testing

Question

To test the `convert` method, what does a test case consist of?

Answer

A `double` and two `int`s that satisfy the precondition.

Question

Which `doubles` do we use?

Testing

Question

Which `doubles` do we use?

Answer

Randomly chosen ones and boundary cases.

Question

What are the boundary cases?

Testing

Question

What are the boundary cases?

Answer

0 and Double.MAX_VALUE.

Question

Which `int`s do we use?

Testing

Question

Which `int`s do we use?

Answer

All combinations of the four constants of the `Converter` class.

Testing

Question

How do we check that

```
double converted = Converter.convert(amount, from, to);
```

is correct?

Testing

Question

How do we check that

```
double converted = Converter.convert(amount, from, to);
```

is correct?

Answer

```
double converted = Converter.convert(amount, from, to);
double doubleConverted =
    Converter.convert(converted, to, from);
final double EPSILON = 0.000001;
boolean correct =
    Math.abs(doubleConverted - amount) < EPSILON;
```

Question

Why do the test cases fail?

Testing

Question

Why do the test cases fail?

Answer

Overflow.

Question

What needs to be fixed, the implementation, the algorithm or the specification?

Testing

Question

What needs to be fixed, the implementation, the algorithm or the specification?

Answer

The specification (since the algorithm and implementation cannot be fixed).

New specification

Problem

Implement this API of the `Converter` class.