Welcome to
Introduction to Computer Science II
EECS 1030

moodle.yorku.ca

Instructor

- Name: Franck van Breugel
- Email: franck@cse.yorku.ca
  Please use your EECS or York account to send me email
- Office: Lassonde Building, room 3046
- Office hours: to be announced

Evaluation

- Week 3: test 1 (2%)
- Week 4: test 2 (20%)
- Week 6: test 3 (22%)
- Week 9: test 4 (22%)
- Week 11: test 5 (20%)
- Other weeks: 7 labs (2% each)

Labs

Students should attend the lab for which they are registered.

- Week 1, 2, 5, 7, 8, 10 and 12 there is a lab (see Moodle for details).
- Each lab is worth 2%.
  - 1% for correctness: 1 if your code passes all the tests of the provided tester, 0 otherwise.
  - 1% for style: 1 if your code passes checkstyle, has proper indentation and descriptive variable names, and is not much more complicated than needed, 0 otherwise.

Checkstyle

To check that the code of your class `X.java` conforms to the code conventions, use

- on a computer in the lab: `checkstyle.sh X.java`
- on your computer: to be announced later

Labs

- Students may do their labs in groups of arbitrary size. Research has shown that groups of two to four students are most effective. You may do the labs on your own.
- Each group is expected to do their own work. If two or more groups collaborate, they should merge into one group, rather than submitting two very similar solutions.
- Advanced software tools will be used to detect the copying of code. If one group copies from another group, both groups are academically dishonest.

Tests

- Week 3, 4, 6, 9 and 11 there is a test in the lab (see Moodle for details).
- Test 2, 3, 4 and 5 will be roughly 75 minutes. Test 1 will be shorter.
- Each test will consist of one or more programming questions and several conceptual questions.
- Students with a documented reason for missing a test, such as illness, compassionate grounds, etc., will have the weight of the missed test(s) shifted to a make up test which will be held during the exam period. This make up test will cover all the material covered in the course.
- During tests, students are expected to do their own work. Looking at someone else's work during the test, talking during the test, using aids not permitted (such as a phone) during the test, and impersonation are all examples of academically dishonest behaviour.

Labs

- Lassonde Building, room 1004 and 1006 (and 1002)
- Lab 1: Wednesdays, 16:00-17:30
  Lab 2: Wednesdays, 13:00-14:30
- The first lab is held this week.
- Tests will be held during the labs.

If you do not have an EECS account yet, activate it

`www.eecs.yorku.ca/activ8`.

Do this before your first lab (although it only takes a few minutes to fill in the form, it may take up to 25 minutes before your account is active).

Drop deadline

March 6

Until this date you can drop the course without getting a grade for it and, hence, it will not affect your gpa.

www.registrar.yorku.ca/enrol/dates/fw14.htm contains important dates.

Academic honesty

"If you put your name on something, then it is your work, unless you explicitly say that it is not."

Examples of academic dishonesty include

- copying code,
- looking at someone else's work during a test,
- talking during a test,
- using aids not permitted (such as a phone) during a test,
- impersonation.

Read http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/ for more details.

Academic honesty

Academic honest behaviour of students increases the value of your degree.

- The instructors will do their best to design tests and policies that promote honest behaviour.
- The students are expected to behave honestly.

Textbook

Franck van Breugel and Hamzeh Roumani. *Introduction to Computer Science II: The Implementer's View*. Only the first 5 chapters are available. More chapters will be provided later in the term.

Study the textbook

The textbook is required for this course.

Studying only the slides and your lecture notes is not sufficient. There will be questions on tests about material that is not covered in class. Therefore, you should study the textbook.

Although you need to memorize some material, most of the material you have to understand.

Students are expected to . . .

- attend the lectures (3 hours per week)
- attend the lab (1.5 hours per week)
- do the lab (on average 1.5 hours per week)
- study the textbook (on average 2 hours per week)
- practice programming (on average 1 hour per week)

How does this course fit within the program?

- EECS 1020: how to use classes
- EECS 1030: how to implement classes
- EECS 2011: how to represent and manipulate data

The programming skills and knowledge of basic concepts that you develop in these three courses will be useful in many other courses.

This course

Problem

Given an API, implement it.

This course

Problem

Given an API, implement it.

Example

Implement this API of the `Converter` class.

Software development

As we have already seen in EECS 1020, the process of software development consists of several phases including

- analysis (define the problem)
- design (solve the problem)
- implementation (write and compile the code)
- testing (run the code)

Analysis

Question

What is the result of the analysis phase?

Analysis

Question

What is the result of the analysis phase?

Answer

A specification.

Analysis

Question

What is the result of the analysis phase?

Answer

A specification.

Question

In our `Converter` example, what is the specification?

Analysis

Question

What is the result of the analysis phase?

Answer

A specification.

Question

In our `Converter` example, what is the specification?

Answer

The API of the `Converter` class.

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

What does it mean for a field (public attribute) to be static?

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

What does it mean for a field (public attribute) to be static?

Answer

It is associated to a class, not to an individual object.

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

What does it mean for a field to be final?

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

What does it mean for a field to be final?

Answer

It is a constant.

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

Why is the constant `CENTIMETER` static, that is, why is it associated with the `Converter` class and not with individual `Converter` objects?

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

Why is the constant `CENTIMETER` static, that is, why is it associated with the `Converter` class and not with individual `Converter` objects?

Answer

If it were static, each `Converter` object would have this constant, which would be a waste of space.

Question

How to declare a public static final attribute?

Question

How to declare a public static final attribute?

Answer

`public static final` *type name* = *value*;

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

Where do you find the value of this constant `CENTIMETER`?

Let's have a look at the API

```
public static final int CENTIMETER
```

Question

Where do you find the value of this constant `CENTIMETER`?

Answer

In the API, go the Field Detail and click on Constant Field Values.

Implementation

Rather than using an editor such as jEdit, a compiler such as javac and an interpreter such as java, most developers use an integrated development environment (IDE).

Popular IDEs for Java are

- Eclipse
- NetBeans

There are many other IDEs for Java.

Clips how to install and use eclipse can be found here.

Documentation

A class contains

- internal documentation
  - one line comments, prefixed by //
  - multi-line comments, surrounded by /* and */
- external comments, surrounded by /** and */

Constructor

Question

Eventhough the code of the `Converter` class does not contain a constructor, when we generate the API it constains a constructor. How is this possible?

Constructor

Question

Eventhough the code of the `Converter` class does not contain a constructor, when we generate the API it constains a constructor. How is this possible?

Answer

If a class does not contain a constructor, the compiler adds a public default constructor, that is, a constructor with no parameters, to the class.

Constructor

Question

How can we make sure that the compiler does not add such a constructor?

Constructor

Question

How can we make sure that the compiler does not add such a constructor?

Answer

By adding a private default constructor. This constructor does not appear in the API, as it is private, and it ensures that the compiler does not add a public default constructor.