Marks for programming exercises

Your marks for the programming exercises can be found at https://www.cse.yorku.ca/~roumani/ePost/server/ep.cgi?year=2014-15&term=F&course=5910

You need to provide your EECS login and password.

The line
CPS weight=5%: max=5: due=September 30, 2014: available=October, 2014
specifies that

- the weight of the programming exercise is 5%,
- the maximal score for the exercise is 5,
- the deadline for the exercise is September 30, 2014, and
- the mark and feedback are available on October 1, 2014.

Feedback for programming exercises

Feedback for your programming exercises is mailed to your EECS account, which you can access at

https://mail.cse.yorku.ca

You need to provide your EECS login and password.

Analysis

Problem

Print on the console

```
Move your mouse immediately after
entering the width of the screen in centimeters:
```

Compute the average speed of the mouse during 0.1 seconds in miles per hour. Print on the console the average speed with two digits precision.

Design

To solve the problem, we can use components that

- return x-coordinate of the mouse cursor
- return y-coordinate of the mouse cursor
- return the maximal x-coordinate (minimum is zero)
- return the maximal y-coordinate (minimum is zero)
- pause the execution by *n* milliseconds

Question

How do we solve the problem?

Components

Each component consists of

- a jar (Java archive) file and
- an API.

To use the component,

- download the jar file and add it to the classpath and
- study the API.

Study the APIs

Study the APIs of

- franck.cse5910.Mouse
- franck.cse5910.Timing

Assertions

```
1  int speed = ...;
2  ...
3  assert speed >= 0;
4  ...
```

According to programmer, whenever we reach line 3, the value of the variable speed is non-negative.

Assertions

```
1 int speed = ...;
2 ...
3 assert speed >= 0;
4 ...
```

According to programmer, whenever we reach line 3, the value of the variable speed is non-negative.

Running your app with assertions enabled (during development)

```
java −ea MouseSpeed
```

Assertions

```
1 int speed = ...;
2 ...
3 assert speed >= 0;
4 ...
```

According to programmer, whenever we reach line 3, the value of the variable speed is non-negative.

Running your app with assertions enabled (during development)

```
java −ea MouseSpeed
```

Running your app without assertions enabled (once deployed)

```
java MouseSpeed
```

Equality

Question

How would you test whether the speed of your mouse and Usain Bolt are the same?

Equality

Question

How would you test whether the speed of your mouse and Usain Bolt are the same?

Answer

```
final double EPSILON = 1.E-5;
boolean equal = Math.abs(mouse - bolt) < EPSILON;
```

Equality

Question

How would you test whether the speed of your mouse and Usain Bolt are the same?

Answer

```
final  double EPSILON = 1.E-5;
boolean equal = Math.abs(mouse - bolt) < EPSILON;
```

Question

Why not simply use boolean equal = (mouse == bolt)?

Equality

Question

How would you test whether the speed of your mouse and Usain Bolt are the same?

Answer

```
final  double EPSILON = 1.E−5;
boolean equal = Math.abs(mouse − bolt) < EPSILON;
```

Question

Why not simply use boolean equal = (mouse == bolt)?

Answer

Because most real numbers are not represented exactly (round-off errors).

To do

- Study Chapter 3 of the textbook.