Homework HW1

The answers you handed in at the end of lecture 1 showed mass confusion! Perhaps 70% of you weren't sure what to write. This was not graded! It was only to help us and you assess the situation.

Doing HW1 will eliminate the confusion. Piazza note @22, (find a link to it in the pinned Piazza Recitation/Homework note.)

Evaluation, Execution, Syntax, Semantics.
Presenting an algorithm in English (2.5 minutes).
Executing the assignment statement (2.5 minutes).
Do HW1 and submit on the CMS

PPT slides, JavaHyperText.

CMS. Visit course webpage, click “Links”, then “CMS for 2110”.
Download ppt slides the evening before each lecture, have them available in class. Please don’t ask questions on the piazza about that material the day before the lecture!
Got a Java question? See first if it’s answered on JavaHyperText

Java OO (Object Orientation)

Python and Matlab have objects and classes.
Strong-typing nature of Java changes how OO is done and how useful it is. Put aside your previous experience with OO (if any).
This lecture:
First: describe objects, detailing their creation and use.
Second: Show you a class definition, a blueprint for objects, and how it contains definitions of methods (functions and procedures) that appear in each object of the class.
Third: Talk about keyword null.
Fourth: Introduce Exceptions

Java OO

References to JavaHyperText entries
- Objects: object
  Calling methods: method call
  Class definition: class def
  public, private: public private method
- Parameter vs argument:
  parameter, argument
  Inside-out rule
- Methods may have parameters
- Method calls may have arguments

Fields of an object may be mentioned. We cover these in next lecture

Function: a method that returns a result.
Procedure: method that does not return a result, void method.

Drawing an object of class javax.swing.JFrame

This object is associated with a window on your computer monitor

Name of object, giving class name and its memory location (hexadecimal 25c7).
Java creates name when it creates object

JFrame@25c7
    hide() show()
    setTitle(String) getTitle()
    getX() getY() setLocation(int, int)
    getWidth() getHeight() setSize(int, int)

Object contains methods (functions and procedures), which can be called to operate on the object

Function: returns a value; call on it is an expression
Procedure: does not return a value; call on it is a statement
**Evaluation of new-expression creates an object**

Evaluation of
\[
\text{new \ javax.swing.JFrame()}
\]
creates an object and gives as its value the name of the object

If evaluation creates this object, value of expression is 
\[
\text{JFrame@25c7}
\]

**A class variable contains the name of an object**

If variable \(h\) contains the name of an object, you can call methods of the object using dot-notation:

Procedure calls: \(h\).show(); \(h\).setTitle("this is a title");

Function calls: \(h\).getX() \(h\).getWidth()

DEMOTIME

**First class definition**

```java
/** An instance (object of the class) has (almost) no methods */
public class C {
}
```

Then, execution of
\[
\text{C k; k = new C();}
\]
creates object shown to right and stores its name in \(k\)

**A class variable contains the name of an object**

If evaluation of \(\text{new \ exp}\) creates the object shown, name of object is stored in \(h\)

```java
javax.swing(JFrame h; h = new javax.swing.JFrame();)
```

**Class extends (is a subclass of) JFrame**

```java
/** An instance is a subclass of JFrame */
public class C extends javax.swing.JFrame {
}
```

Object has 2 partitions: one for JFrame methods, one for C methods

Easy re-use of program part!
Class definition with a function definition

```java
/** An instance is a subclass of JFrame with a function area */
public class C extends javax.swing.JFrame {
    /** Return area of window */
    public int area() {
        return getWidth() * getHeight();
    }
}
```

Inside-out rule for finding declaration

```java
/** An instance is a JFrame with more methods */
public class C extends javax.swing.JFrame {
    /** Return area of window */
    public int area() {
        return getWidth() * getHeight();
    }
    public void setWtoH() {
        setSize(getHeight(), getHeight());
    }
    public void setTitleToDate() {
        setTitle(new java.util.Date().toString());
    }
}
```

Using an object of class Date

```java
/** An instance is a JFrame with more methods */
public class C extends javax.swing.JFrame {
    /** Put the date and time in the title */
    public void setTitleToDate() {
        setTitle(new java.util.Date().toString());
    }
}
```

About null

```java
v1 = C@16;

v2 = null;

null denotes the absence of a name.

v2.getName() is a mistake! Program stops with a NullPointerException.

You can write assignments like: v1 = null;

and expressions like: v1 == null
When the assert statement is executed and x is not 6, an object of class AssertionError is created and “thrown”. It contains info needed to print out a nice message.

When 5/0 is evaluated, an object of class ArithmeticException is created and “thrown”. It contains info needed to print out a nice message.

You will learn all about exceptions in next week’s recitation!