Piazza: Q&A platform for students

1. Goto www.piazza.com
2. “Student? Search your class”
3. “Search Schools:” University of Maryland (umd.edu)
4. “Class *X*:” CMSC 131 0101: Introduction to Object Oriented Programming
   – *Not other 131 options with “Plane” in the title*

Piazza: Q&A platform for students

5. “Join as Student”
6. “Add Classes”
7. Login if you already have an account, or enter your email address @umd.edu
8. You will receive a confirmation email with a link
9. Click the link to activate your account

Eclipse

Integrated **D**evelopment **E**nvironment

A GUI that makes development easier for the programmer

- Syntax highlighting
- Autocompletion
- Compile and run with the click of a button

Eclipse

- Eclipse has several modes called "Perspectives" used for different tasks
  - "Java" perspective
  - "CVS repository exploring" perspective
- Switching perspectives:
  - Through Menu:

    Window->Open perspective(->Other)
  - Or through buttons at top right

Creating your own projects

1. From main menu:
   - File->New->Java Project
   - Enter a name and click “Finish”
2. In package explorer tab on left (Java perspective)
   - Right click Project folder: New->Class
   - Enter a name
   - For “which method stubs would you like to create?” check “public static void main(String[] args)
   - Click “Finish”

Accessing class projects

- In “CVS Repository Exploring” perspective:
  - Right-click in “CVS Repositories” tab on left:

    New->Repository location
  - Fill out dialog box as shown in class website resource page
    - Check save password so you don’t need to enter it every time
  - Expand directory tree in “CVS Repositories” tab
  - Find project folder under “HEAD”
  - Right-click on project folder -> “Check out”

Accessing class projects

- In “Java” perspective:
  - The checked-out project will now appear as a folder in the “Package Explorer” tab.
  - Expand the project folder: Project folder -> src -> default package
  - Double-click on the .java file to edit it
  - Click the green arrow button in the toolbar to run your code.
  - Every time you save, a backup of your code will be saved in the CVS repository.

Submitting class projects

- In “Java” perspective:
  - In “Package Explorer” tab, right-click on top-level folder for project
  - Select “Submit Project” (should appear after “Debug As”).
  - Enter UMD directoryID and password
  - Project will be uploaded to the Submit Server

Submit server

- CS Department's system for automated testing of student projects
- https://submit.cs.umd.edu/summer2013
- Log on with your UMD directoryID and password
- Click link for the class
- You'll be able to view a history of your submissions and test results for each lab and project (Lab01 has no automated tests)

Web submission

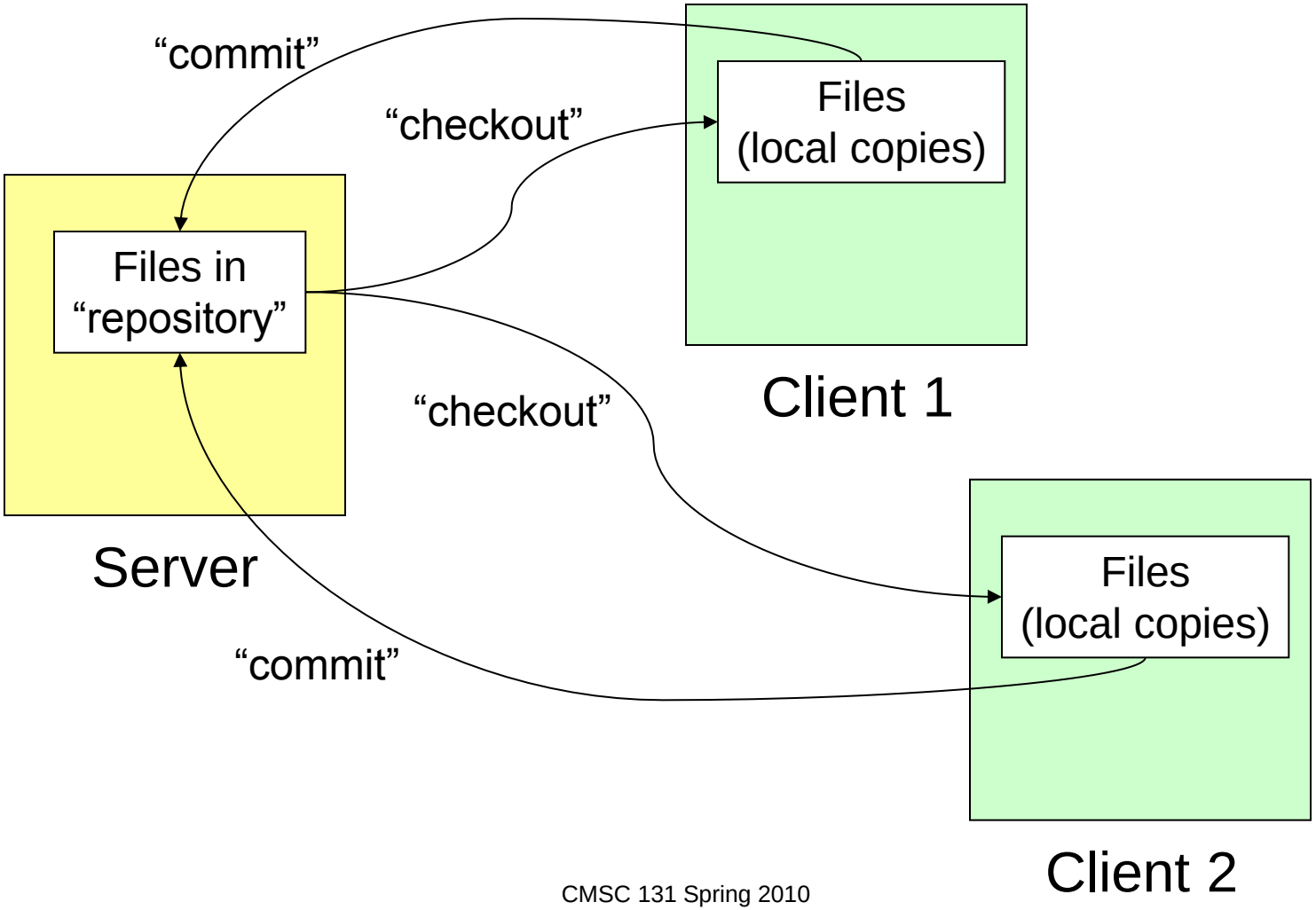
- For cases where Eclipse submission is not working
- On the submit server:
  - In the “Projects” section, find the row for the project you’d like to submit
  - Click on “submit” link in “Web submission” column
  - You will be able to upload your project files

CVS Repository

Concurrent **V**ersioning **S**ystem

- A system that maintains multiple versions of multiple files from multiple developers
- Useful for large developer teams working on large software projects
- We'll just use it to keep all of your work backed up in a centralized location
- Separate from the submit server

CVS Repository



Operators and Expressions

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html

What is an operator?

$$3 + 7 = 10$$

op(in1, in2, in3, …) = value

- A function that maps several inputs to a single output value
- + is a binary operator: It takes two inputs
- A “unary” operator takes one input
- A “ternary” operator takes three inputs
- An “N-ary” operator takes N inputs

Some familiar operators

- Arithmetic
  - Plus (+)
  - Minus (-)
  - Multiply (*)
  - Divide (/)
- An example of an arithmetic unary operator?

Arithmetic in Java

- Adding bytes and shorts

```
short x = 3;
short y = 4;
short z = x+y; // What happens?
```

- Integer division

```
int x = 3;
int y = 4;
int z = x/y; // What happens?
```

Arithmetic in Java

- Adding bytes and shorts

```
short x = 3;
short y = 4;
short z = (short)(x+y); // int result
```

- Integer division

```
int x = 3;
int y = 4;
int z = x/y; // z is now 0
```

Arithmetic in Java

- Modulus (%)
  - Fancy word for remainder

```
int x = 3;
int y = 4;
int z = x%y; // What is z?
```

- How do these operators interact with floating points?

Logical operators

- && - Logical AND
- || - Logical OR
- ! – Logical negation
- Logical XOR: ????

Logical operators

- && - Logical AND
  - Each input is a “conjunct”
- || - Logical OR
  - Each input is a “disjunct”
- ! – Logical negation
- Logical XOR: (A || B) && !(A && B)

Relational operators

- Logical equality: ==
  - Don’t confuse with assignment! (Single ‘=‘ sign)
- Logical inequality: !=
- Logical comparison: <, <=, >=, >

Unary

- Arithmetic negation (-)
- Logical negation (!)
- Increment (++) and Decrement (--)
  - Prefix: int x = 3; x++; //Now x is 4
  - Postfix: int x = 3; ++x; //Now x is 4?
- What's the difference?

Unary

- Arithmetic negation (-)
- Logical negation (!)
- Increment (++) and Decrement (--)
  - Postfix: int i = 3; int j = i++; //i is 4, j is 3
  - Prefix: int i = 3; int j = ++i; //i is 4, j is 4
- What's the difference?
  - Process from left to right:
    - x++: The output value is x before incrementing. The input variable, x, is incremented.
    - ++x: The input variable, x, is incremented. The output value is x after incrementing.

Logical short-circuit

- If A == false, then A && B == ??
- If A == true, then A && B == ??
- If A == true, then A || B == ??
- If A == false, then A || B == ??

Logical short-circuit

- If A == false, then A && B == false
- If A == true, then A && B == B
- If A == true, then A || B == true
- If A == false, then A || B == B
- Sometimes only the first conjunct/disjunct need be evaluated. In this case, Java ignores the second.

Assignment operator

- Assignment is an operator??
- What are the inputs?
- What is the output?
- Arithmetic and assignments:
    - +=
    - -=
    - *=
    - /=
    - %=

Assignment operator

- Assignment is an operator??
- What are the inputs? The data on either side
- What is the output? The assigned value
- Arithmetic and assignments:
    - +=
    - -=
    - *=
    - /=
    - %=

Order of operations

- Some operators get evaluated before others. This is called precedence.
- Like PEMDAS for arithmetic.
- For equal precedence, operators are applied left to right.
- What is 5-3-3?
- What is ++i++?

“Bit-wise” logical operators

- & (and)
- | (or)
- ^ (xor)
- ~ (complement)
- For integer operands, these are applied bit by bit.
- &, |, and ^ also work on booleans, but *don’t* short circuit.

http://docs.oracle.com/javase/specs/jls/se7/html/jls-15.html#jls-15.22

“Bit-wise” logical operators

0011	0011	0011
& 0101	| 0101	^ 0101
????	????	????

~ 0101	~ 0011
????	????

“Bit-wise” logical operators

0011	0011	0011
& 0101	| 0101	^ 0101
0001	0111	0110

~ 0101	~ 0011
1010	1100

“Bit-wise” shift operators

- Shift the bits of a number *n*
- *s* is the size of the shift
- Shift left (feed in zeros): *n* << *s*

  n: 01101101

  n<<1: 11011010

  n<<2: 10110100
- Shift right (preserve sign): *n* >> *s*
- Shift right (feed in zeros): *n* >>> *s*

Bitwise shift operators

What do the following represent in terms of / and %?

- n << s == ??
- n >> s == ??
- 1 << s == ??
- n & 1 == ??

Bitwise shift operators

What do the following represent in terms of *, /, and %?

- $n << s == n * 2^s$
- $n >> s == n / 2^s$
- $1 << s == 2^s$
- $n \& 1 == n \% 2$

Operators as “pre-methods”

- A + B
- plus(A,B)
- op(A,B,C,….)
- Methods: operators written by you
  - Factorial()
  - System.out.println();
  - Intertwined with objects

Coming attractions

- Control flow and loops
- Quiz 1 on Thursday
- Project 0 next week