CSE 331: Developer Tools

Section 2
10/4/2012

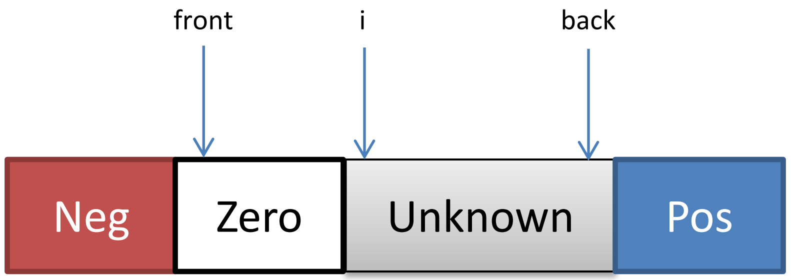
Slides by: Kellen Donohue
with material from Krysta Yousoufian

Agenda

- Loop development & ex0
- Tools
  - Eclipse
  - Subversion
  - JUnit

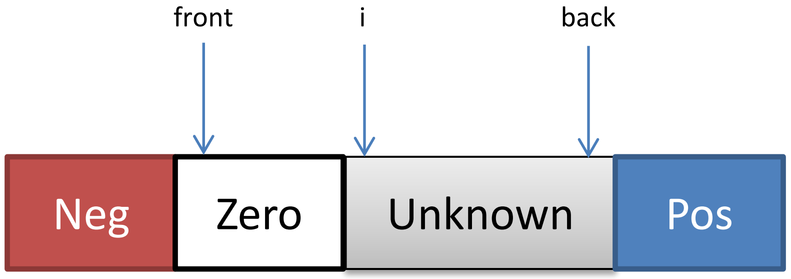
What’s wrong?

```
public static void partition(int[] b) {
    int frontIndex = 0;
    int backIndex = b.length - 1;
    for (int i = 0; i <= backIndex; i++) {
      if (b[i] < 0) {
        swap(b, frontIndex, i);
        frontIndex++;
      } else if (b[i] > 0) {
        swap(b, backIndex, i);
        backIndex--;
      }
    }
  }
```



What’s wrong?

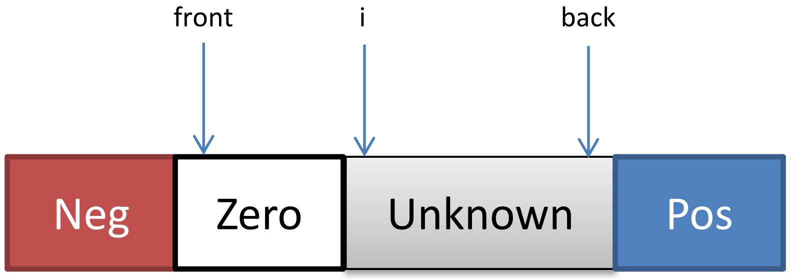
```
public static void partition(int[] b) {
    int frontIndex = 0;
    int backIndex = b.length - 1;
    for (int i = 0; i <= backIndex; i++) {
      if (b[i] < 0) {
        swap(b, frontIndex, i);
        frontIndex++;
      } else if (b[i] > 0) {
        swap(b, backIndex, i);
        backIndex--;
      }
    }
}
```



[0, -1, 2, -3]

What’s wrong?

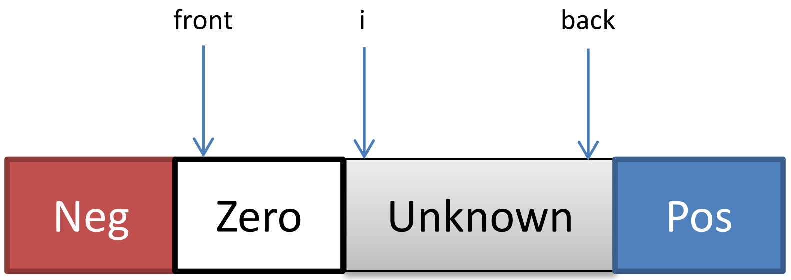
```
public static void partition(int[] b) {
    int frontIndex = 0;
    int backIndex = b.length - 1;
    for (int i = 0; i <= backIndex; i++) {
      if (b[i] < 0) {
        swap(b, frontIndex, i);
        frontIndex++;
      } else if (b[i] > 0) {
        swap(b, backIndex, i);
        backIndex--;
      }
    }
  }
```



[0, -1, 2, -3] => [-1, 0, 2, -3]

What’s wrong?

```
public static void partition(int[] b) {
    int frontIndex = 0;
    int backIndex = b.length - 1;
    for (int i = 0; i <= backIndex; i++) {
      if (b[i] < 0) {
        swap(b, frontIndex, i);
        frontIndex++;
      } else if (b[i] > 0) {
        swap(b, backIndex, i);
        backIndex--;
      }
    }
  }
```



[0, -1, 2, -3] => [-1, 0, 2, -3] => [-1, 0, -3, 2]

What’s wrong?

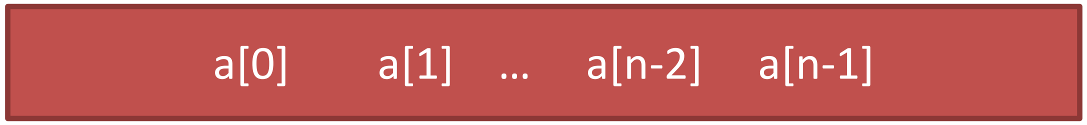
```
public static void partition(int[] b) {
    int frontIndex = 0;
    int backIndex = b.length - 1;
    for (int i = 0; i <= backIndex; i++) {
      if (b[i] < 0) {
        swap(b, frontIndex, i);
        frontIndex++;
      } else if (b[i] > 0) {
        swap(b, backIndex, i);
        backIndex--;
        i--;
      }
    }
  }
```

Loop development example

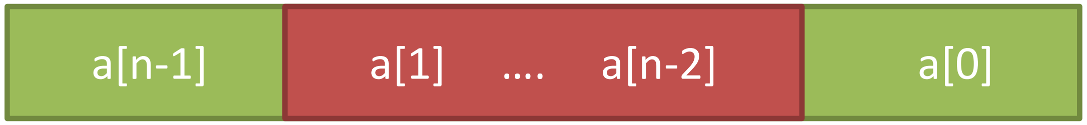
- Given array a = [0, …, n-1], reverse the elements in a
- pre: a[0] a[1] … a[n-2] a[n-1]
- post: a[n-1] a[n-2] … a[1] a[0]

Loop development example

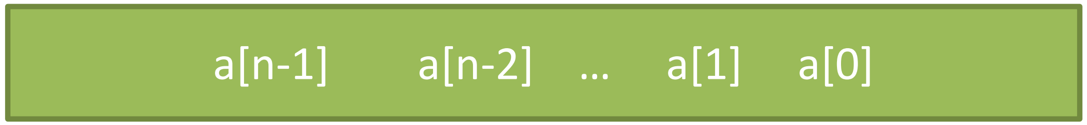
- Given array a = [0, …, n-1], reverse the elements in a
- pre:



- loop-inv:

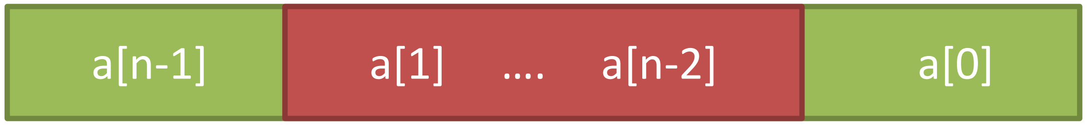


- post:



Loop development example

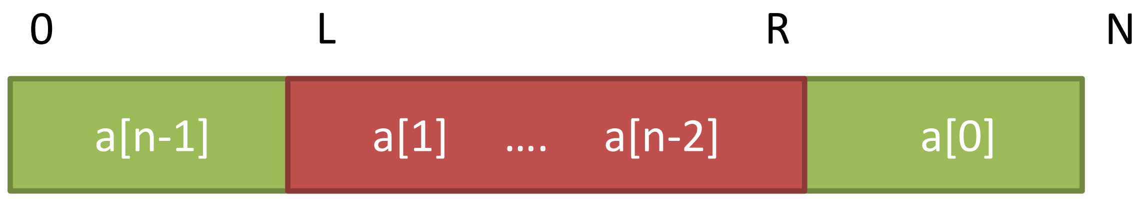
- loop-inv:



```
L = 0;
R = n-1;
while (L < R) {
      swap(a[L],a[R]);
      L = L+1;
      R = R-1;
}
```

Loop development example

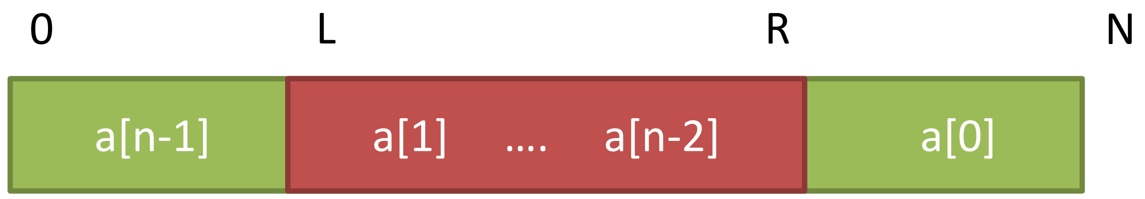
- loop-inv:



```
L = 0;
R = n-1;
while (L < R) {
      swap(a[L],a[R]);
      L = L+1;
      R = R-1;
}
```

Loop development example

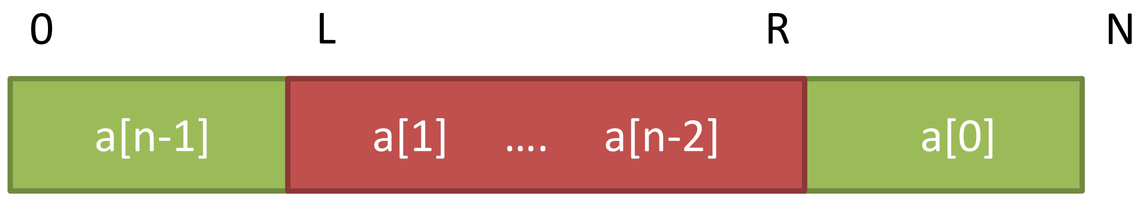
- loop-inv:



```
[0..L-1] and [R+1..n-1] are reversed, rest
normal

L = 0;
R = n-1;
while (L < R) {
      swap(a[L],a[R]);
      L = L+1;
      R = R-1;
}
```

Loop development example

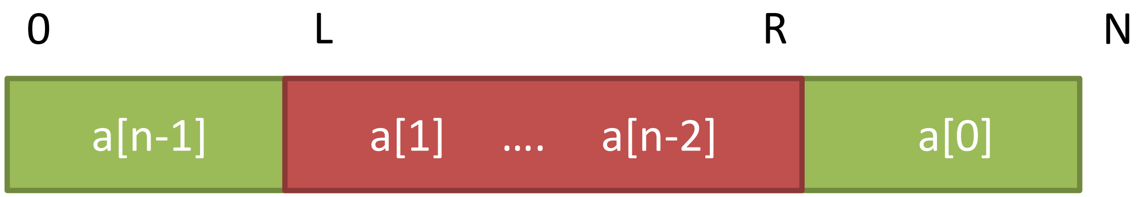
- loop-inv:



```
[0..L-1] and [R+1..n-1] are reversed, rest
normal

L = 0;
R = n-1;                   // I) True before loop
while (L < R) {
     swap(a[L],a[R]);
     L = L+1;
     R = R-1;
}
```

Loop development example

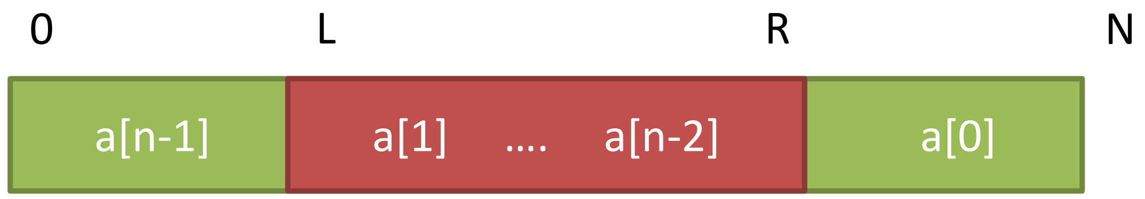
- loop-inv:



```
[0..L-1] and [R+1..n-1] are reversed, rest
normal

L = 0;
R = n-1;                    // I) True before loop
while (L < R) {
      swap(a[L],a[R]);
      L = L+1;
      R = R-1;             // II) True inductively
}
```

Loop development example

- loop-inv:



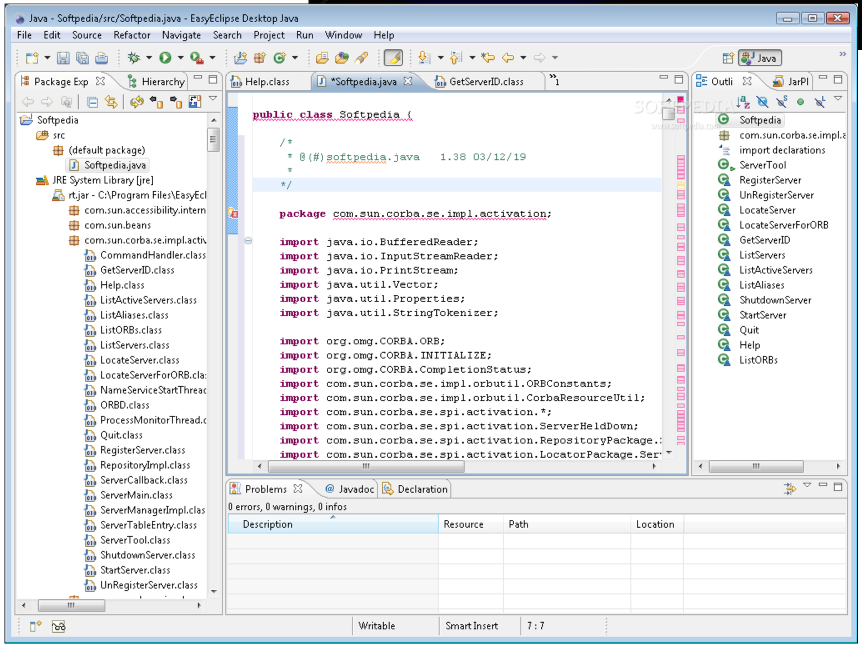
```
[0..L-1] and [R+1..n-1] are reversed, rest
normal

L = 0;
R = n-1;                    // I) True before loop
while (L < R) {
     swap(a[L],a[R]);
     L = L+1;
     R = R-1;              // II) True inductively
}                          // III) True after loop
```

Agenda

- Loop development on ex0
- Tools
  - Eclipse
  - Subversion
  - JUnit

Eclipse

- Eclipse is a multi-platform, open-source IDE
- Build, edit, run, test, distribute your code from one program



```java
public class Softpedia {

    /*
     * @(#)softpedia.java   1.38 03/12/19
     *
    */

    package com.sun.corba.se.impl.activation;

    import java.io.BufferedReader;
    import java.io.InputStreamReader;
    import java.io.PrintStream;
    import java.util.Vector;
    import java.util.Properties;
    import java.util.StringTokenizer;

    import org.omg.CORBA.ORB;
    import org.omg.CORBA.INITIALIZE;
    import org.omg.CORBA.CompletionStatus;
    import com.sun.corba.se.impl.orbutil.ORBConstants;
    import com.sun.corba.se.impl.orbutil.CorbaResourceUtil;
    import com.sun.corba.se.spi.activation.*;
    import com.sun.corba.se.spi.activation.ServerHeldDown;
    import com.sun.corba.se.spi.activation.RepositoryPackage.:
    import com.sun.corba.se.spi.activation.LocatorPackage.Ser
```

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

Nice features of Eclipse

- Code generation
- Easy refactoring/renaming
- Helpful autocomplete
- Easily see relevant documentation
- Quickly find variable uses/definitions
- Debugging
- Good integration with other tools

Demo

Getting Eclipse

- It's already installed on CSE Lab Machines
  - Open a terminal – type `eclipse &`

- Working from home (instructions in tools handout)
  1. Download Java JDK (Version 7)
  2. Set JAVA_HOME environment variable
  3. Download Eclipse

Alternatives

- Other IDEs: jEdit, Netbeans
- vim / Emacs / gedit / Notepad++ / Textmate
  & command line
- If you’ve only used one environment before – try Eclipse
- Course staff will support Eclipse – something else and you’re (more) on your own

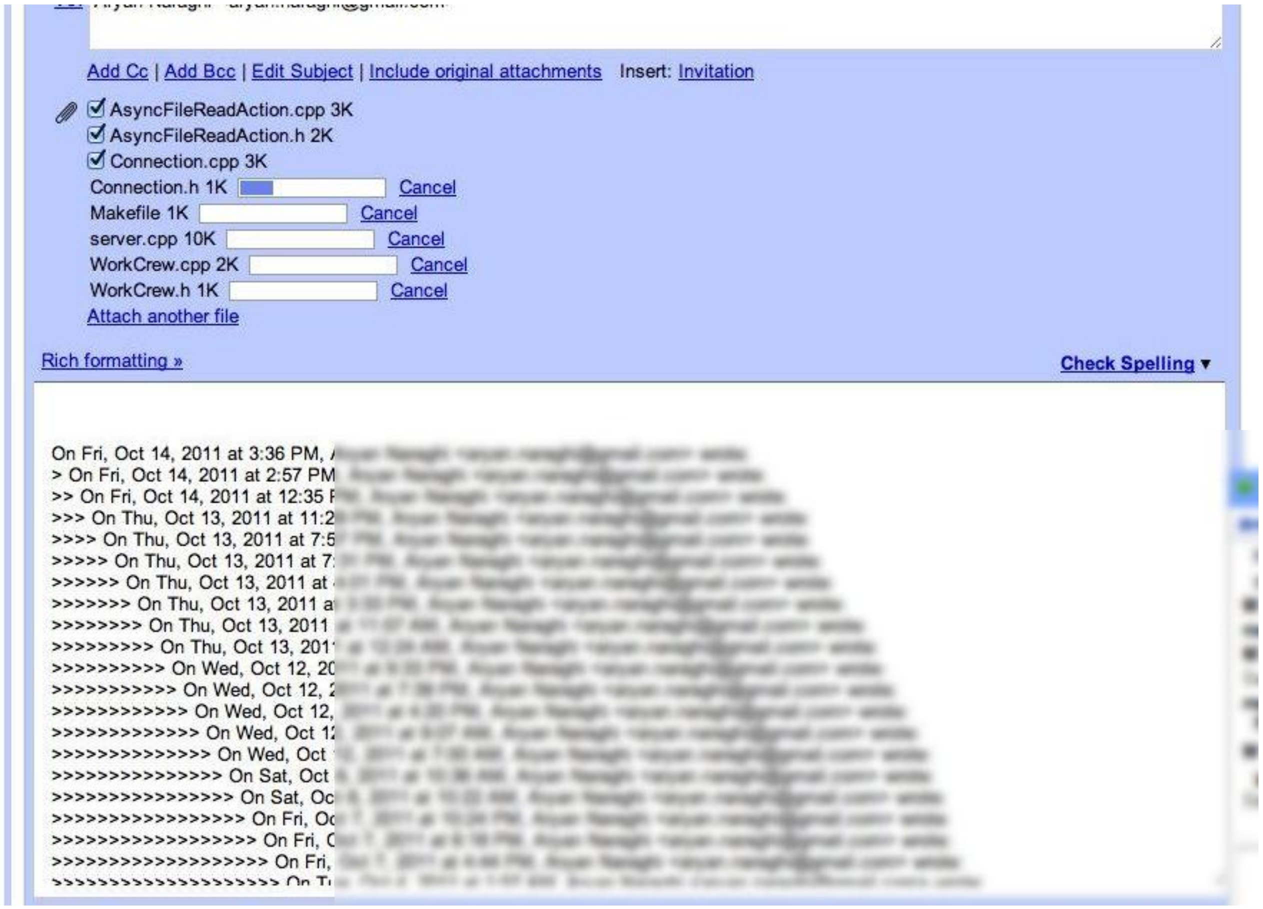
Version Control

- System for tracking changes to code
  - Essential for managing big projects
  - Learn it now – you WILL use it again and again!

- Makes it easy to:
  - See a history of changes
  - Revert back to an older version of your code
  - Back up your work
  - Work on code in a team
  - Work on different machines

- You'll use Subversion (SVN) this quarter
  - There are others: Mercurial, Git, CVS, …

Don’t be this guy



Aryan Naraghi <aryan.naraghi@gmail.com>

Add Cc | Add Bcc | Edit Subject | Include original attachments Insert: Invitation

- ☑ AsyncFileReadAction.cpp 3K
- ☑ AsyncFileReadAction.h 2K
- ☑ Connection.cpp 3K
- Connection.h 1K Cancel
- Makefile 1K Cancel
- server.cpp 10K Cancel
- WorkCrew.cpp 2K Cancel
- WorkCrew.h 1K Cancel

Attach another file

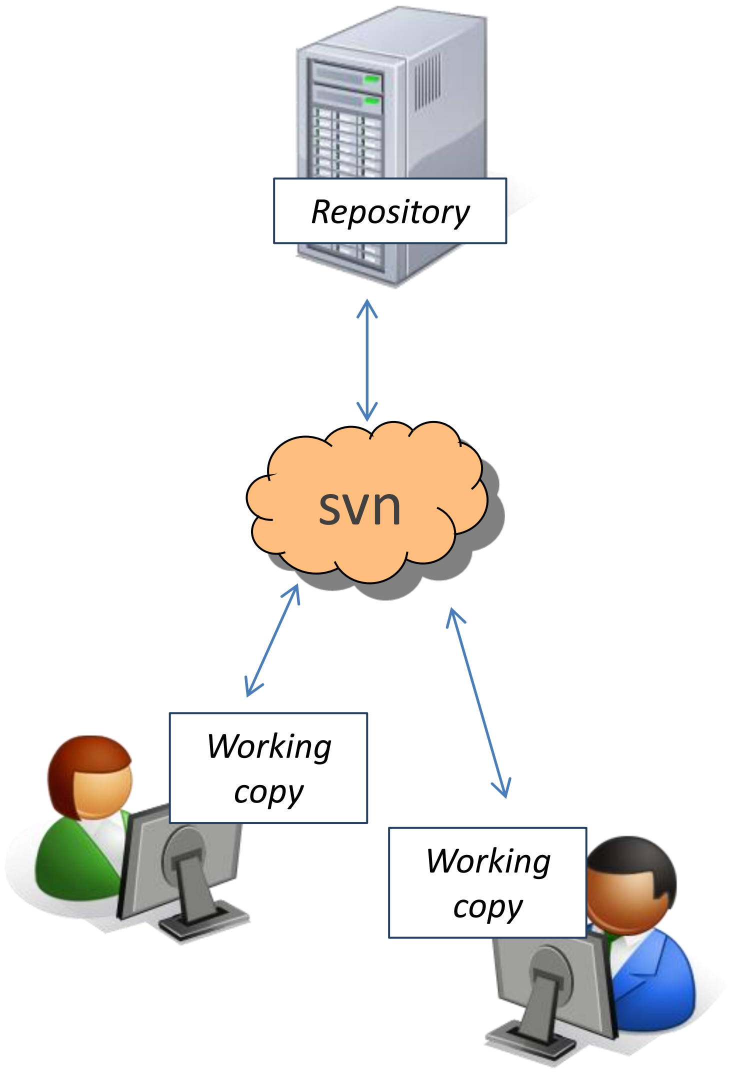
Rich formatting » Check Spelling ▾

```
On Fri, Oct 14, 2011 at 3:36 PM, A
> On Fri, Oct 14, 2011 at 2:57 PM
>> On Fri, Oct 14, 2011 at 12:35 P
>>> On Thu, Oct 13, 2011 at 11:2
>>>> On Thu, Oct 13, 2011 at 7:5
>>>>> On Thu, Oct 13, 2011 at 7:
>>>>>> On Thu, Oct 13, 2011 at
>>>>>>> On Thu, Oct 13, 2011 a
>>>>>>>> On Thu, Oct 13, 2011
>>>>>>>>> On Thu, Oct 13, 201
>>>>>>>>>> On Wed, Oct 12, 20
>>>>>>>>>>> On Wed, Oct 12, 2
>>>>>>>>>>>> On Wed, Oct 12,
>>>>>>>>>>>>> On Wed, Oct 12
>>>>>>>>>>>>>> On Wed, Oct
>>>>>>>>>>>>>>> On Sat, Oct
>>>>>>>>>>>>>>>> On Sat, Oc
>>>>>>>>>>>>>>>>> On Fri, Oc
>>>>>>>>>>>>>>>>>> On Fri, C
>>>>>>>>>>>>>>>>>>> On Fri,
>>>>>>>>>>>>>>>>>>>> On Tu
```

[redacted]

Organization

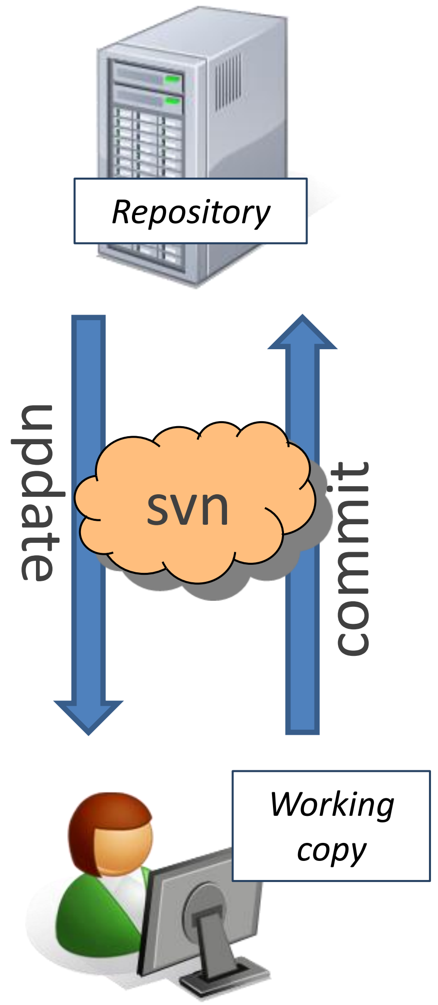
- A *repository* stores the master copy of the project
  - Someone creates the repo for a new project
  - Then nobody touches this copy directly
  - Lives on a server everyone can access
- Each person checks out their own working copy
  - Makes a local copy of the repo
  - You'll always work off of this copy
  - The version control system syncs the repo and working copy



Common Actions

Most common commands:

- Commit / checkin
  - integrate changes *from* your working copy *into* the repository
- Update
  - integrate changes *into* your working copy *from* the repository

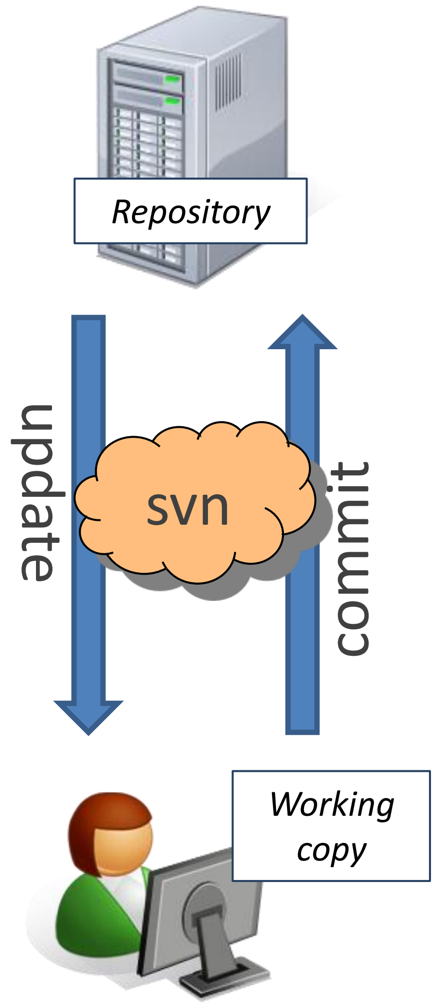


Common Actions

Most common commands:

- Add, delete
  - add or delete a file in the repository
  - just putting a new file in your working copy **does not add it to the repo**

Dropbox is a similar idea, but it adds every file, does commits for every change, and pulls anytime a file is changed elsewhere



This Quarter

- Use Subversion for your homework assignments
- We distribute starter code by adding it to your repo
- You turn in your files by **adding** them to the repo and **committing** your changes
- Run validator tool to make sure you added everything correctly, etc.
- See the version control handout:

  http://www.cs.washington.edu/education/courses/cse331/1au/tools/versioncontrol.html

How to use SVN

- Command line
  - `svn help` – List commands
  - `svn help checkout` – Options for checkout

- Subclipse Plugin for Eclipse

- GUI interfaces -- TortoiseSVN

Subclipse Demo

JUnit

- You wrote a lot of code in Eclipse, and committed it all in Subversion – but does it work?
  - And will it work tomorrow?
  - If there’s a bug how do we know it’s fixed?
  - If something else changes will our code break?

- Unit tests can assuage these fears

- JUnit is a unit-testing framework for Java we will use extensively this quarter

A JUnit test class

```
import org.junit.*;
import static org.junit.Assert.*;

public class PointTest {
    ...

    @Test
    public void testDistance() { // a test case method
        ...
    }
}
```

A method with `@Test` is flagged as a JUnit test case.

All `@Test` methods run when JUnit runs your test class.

Verifying Behavior with Assertions

- Assertions: special JUnit methods
- Verifies that a value matches expectations

```
assertEquals(42, meaningOfLife());      ← fails if meaningOfLife() != 42
assertTrue(list.isEmpty());             ← fails if list.isEmpty() is false
```

- If the value isn't what it should be, the test fails
  - Test immediately terminates
  - Other tests in the test class are still run as normal
  - Results show details of failed tests

Using Assertions

`assertTrue(`**test**`)`	fails if the boolean test is `false`
`assertFalse(`**test**`)`	fails if the boolean test is `true`
`assertEquals(`**expected**`, `**actual**`)`	fails if the values are not equal
`assertSame(`**expected**`, `**actual**`)`	fails if the values are not the same (by ==)
`assertNotSame(`**expected**`, `**actual**`)`	fails if the values *are* the same (by ==)
`assertNull(`**value**`)`	fails if the given value is *not* `null`
`assertNotNull(`**value**`)`	fails if the given value is `null`

- And others: http://www.junit.org/apidocs/org/junit/Assert.html
- Each method can also be passed a string to display if it fails:
  - e.g. `assertEquals("`**message**`", `**expected**`, `**actual**`)`

Checking for Exceptions

- Verify that a method throws an exception
- Place above method:

```
@Test(expected=IllegalArgumentException.class)
```

- Test passes if specified exception is thrown, fails otherwise

- Only time it's OK to write a test with no `asserts`!

```
// Try to access the first item in an empty ArrayList
@Test(expected=IndexOutOfBoundsException.class)
 public void test() {
    List<String> list = new ArrayList<String>();
    list.get(0);
 }
```

Setup and Teardown

- Methods to run before/after each test case method is called:

```
 @Before
public void name() { ... }
@After
public void name() { ... }
```

- Methods to run once before/after the entire test class runs:

```
@BeforeClass
public static void name() { ... }
@AfterClass
public static void name() { ... }
```

JUnit and Eclipse

- Eclipse can easily run JUnit tests and report results.
- This is when the Eclipse debugger is especially helpful!
- Demo

Putting it all together

- HW3 out later today or tomorrow
  - Mostly environment setup & introduction
  - Uses all tools described here
  - Tools handouts on website soon
  - If you get stuck, ask for help!
    - Message board