

Mat 2170 WEEK 1

Drs. N. Van Cleave & A. Mertz

Spring 2010

1

Mat2170 Course Goals

- ▶ Develop **Algorithm Design Skills**:
writing step-by-step instructions to solve problems
- ▶ Develop Facility with the **Object Oriented Paradigm**:
using, extending, and developing Classes and Objects
- ▶ Learn a Subset of the **Syntax of the Java language**:
be capable of writing significant Java programs
- ▶ Develop **Critical Thinking Skills**:
the processes of discernment, analysis and evaluation of information

2

General Course Guidelines

- ▶ Syllabus
- ▶ Schedule (note **evening** exams)
- ▶ Academic Integrity
- ▶ Labs – weekly
- ▶ Quizzes – weekly (Van Cleave)
- ▶ Course Web Site (www.eiu.edu/~mathcs)

3

Week 1 Student Responsibilities

- ▶ **Reading**: Textbook, Chapters 1 and 2.1
- ▶ **Lab**: Lab 1, Prelab, & Postlab
- ▶ **Submissions**: electronic, printouts, and web publishing
- ▶ **Attendance**: lecture & lab

4

Week 1 Topics

- ▶ Getting ready for Lab 1:
 - ▶ Algorithms
 - ▶ The Programming Process
 - ▶ Java
 - ▶ The Mathematics and Computer Science lab
 - ▶ Hello World Program
- ▶ What is Computer Science?
- ▶ Computer Hardware

5

Algorithms

- ▶ An **Algorithm** is a set of instructions for solving a problem — much like a recipe for a particular dish, or the instructions for putting together a model airplane.
- ▶ **Algorithmic Properties**
 - ▶ A **Step-by-step method** for solving a problem
 - ▶ All steps must be **unambiguous** and **executable**
 - ▶ Must **terminate** with the **correct outcome**

6

The Programming Process

► Algorithmic Design

- **Specifications** – types and restrictions of all required input and output for the program
- **Test Suite** – well-selected inputs with **expected** outputs
- **Logic** which solves problem (human readable)
 - General (Outline)
 - Detailed

► Software

- **Coding** – translating Detailed algorithm into computer language (JAVA)
- **Debugging** – locating and eliminating errors
- **Maintenance** – evolution of program over time

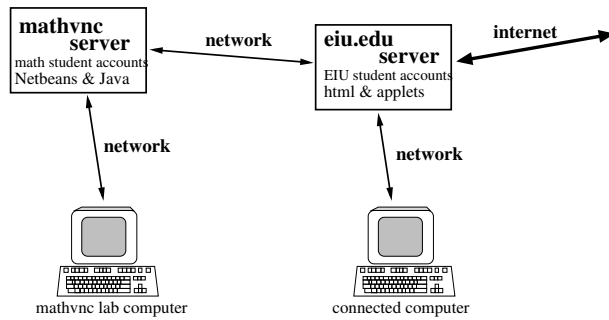
7

What you'll need in lab to program in Java:

1. An account on the **mathvnc** server, issued in class
2. Your **userid** (same as EIU email id) and **password** (change.me)
3. A **terminal window**, from which we can direct commands to the computer system
4. The ACM graphics library: **acm.jar**
5. A way to create the programs with an editor: **netbeans**
6. The java interpreter: **JDK**
7. A web browser: **firefox**
8. A way to transfer files from your account on the mathvnc server to your EIU student account: **websync**

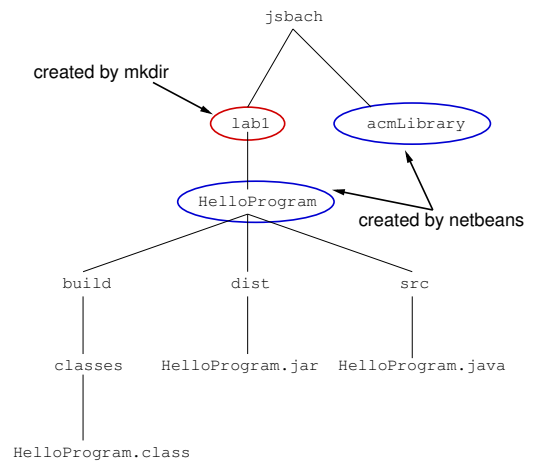
8

The mathvnc and EIU servers



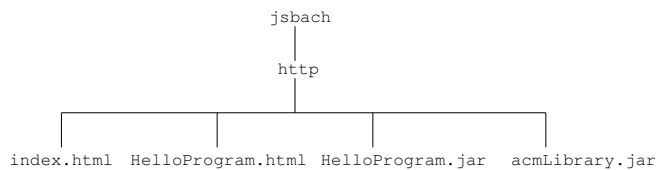
9

Directory Structure in Linux



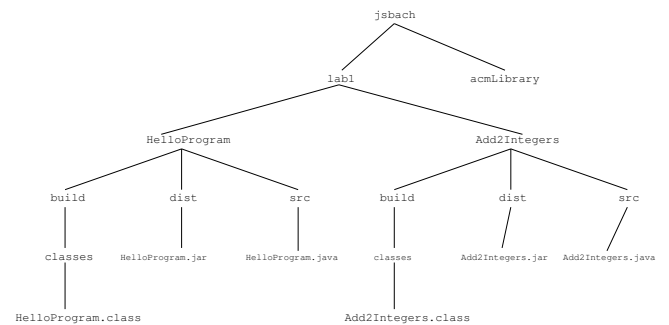
10

Directory Structure in Your EIU Account



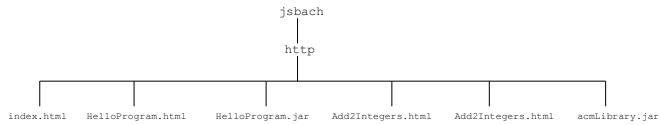
11

Directory Structure after Lab 1 Completed



12

Directory Structure in EIU Account after Lab 1



13

A Few Common Linux Commands

pwd	Display current path
pwd	<i>Shows where you are in file structure</i>
ls	List directory contents
ls	<i>list contents of current directory</i>
ls -l	<i>long version, gives more info</i>
mkdir	Make Directory
mkdir acmLibrary	<i>create the directory acmLibrary in the current directory</i>

14

cd	Change Directory
cd ~	<i>go to home directory</i>
cd ~/lab1	<i>go to home, then down to the lab1 directory</i>
cd lab1	<i>go down into the lab1 directory (does not work if not in home dir)</i>
mv	Move or rename a file
mv fileF ToHere	<i>move fileF into directory ToHere</i>
mv FileA FileB	<i>rename FileA as FileB</i>
passwd	Change mathvnc account password
passwd	<i>You will be prompted for old and new passwords</i>

15

scp	Secure file copy
scp * userid@pen.eiu.edu:~/http	<i>copy all files in current directory to http directory in account userid on pen</i>
	<i>replaced by: websync</i>
	Other Software
firefox	web browser
firefox &	<i>starts the firefox web browser</i>
firefox file.html &	<i>open file.html in firefox</i>
netbeans	IDE for Java
netbeans &	<i>starts the Java IDE</i>

16

2170 programming and acm.jar

- ▶ The **Association for Computing Machinery** provides free java libraries (contained in acm.jar), which we will be using this semester.
- ▶ Among other things, this library supports graphics, graphical user interfaces, and event-driven programming.
- ▶ We can write programs that are more interesting and fun by extending what others have written.
- ▶ Much more information is available at jtf.acm.org

17

The Integrated Development Environment (IDE)

- ▶ An IDE provides an organized way to view and select files from a project, to edit files, and to compile and run programs.
- ▶ There are multiple choices for an IDE — we are going to use netbeans
- ▶ netbeans is freely available from Sun Microsystems
- ▶ To start up netbeans from a terminal window, enter netbeans &
- ▶ netbeans has several windows and menus — follow the lab handout carefully.

18

The Java Developers Kit (JDK)

- ▶ Provides the Java compiler, which netbeans accesses.
- ▶ JDK is freely available from Sun Microsystems
- ▶ We are currently using version JDK 1.6
- ▶ Programs are compiled into the “machine language” of the Java Virtual Machine (JVM).
- ▶ Java then interprets those programs by simulating the JVM.

19

The Hello World Program

```
import acm.graphics.*;
import acm.program.*;

public class HelloProgram
    extends GraphicsProgram
{
    public void run()
    {
        add(new GLabel("hello, world", 100, 75));
    }
}
```

20

Programming Errors and Debugging

- ▶ **Syntax error**
 - ▶ Violation of the grammatical rules of a language
 - ▶ Compiler displays error message(s)
 - ▶ Corrected by tracking error down and editing the program file
- ▶ **Logic/semantic error:**
 - ▶ Sometimes called a **bug**; process of eliminating such errors is called **debugging**
 - ▶ Logic errors are much harder to find and eliminate than syntax errors
 - ▶ Good design and testing is essential to writing robust software
 - ▶ Time spent on design is well worth it

21

Your Web Page

- ▶ EIU provides students with email and a web page.
- ▶ When we compile and run a Java program from netbeans, we call that version an **application**.
- ▶ Java also makes it possible to write small interactive programs, called **applets**, that run under the control of a web browser.
- ▶ Part of your responsibility for labs and programming assignments will be to update your web page to include applets for each program.
- ▶ Follow carefully the instructions given in this week's lab.

22

Creating a Java Applet — Steps taken by the applet author

- ▶ Design and write the Java code: `HelloProgram.java`
- ▶ Generate the intermediate code for the applet: `HelloProgram.jar`
- ▶ Create a small HTML file which tags the applet: `HelloProgram.html`
- ▶ Publish an HTML web page that includes a reference to the compiled applet: `index.html`

23

Running a Java Applet — Steps taken by the applet user

- ▶ Start a browser, enter the URL for the applet page or click on the link to the applet
- ▶ Browser reads and interprets the HTML source for the web page
- ▶ The object html tag causes the browser to download the compiled applet over the network
- ▶ The applet intermediate code is checked to verify it does not violate the security of the user's system
- ▶ The Java interpreter in the browser program runs the compiled applet, which generates the desired display on the user's console

24

Software Maintenance

- ▶ Between 80% and 90% of total software cost is for maintenance **after** it has been released
- ▶ Reasons software requires maintenance:
 - ▶ Continued debugging over time
 - ▶ feature enhancement - updates requested by users or to compete in the marketplace
- ▶ Two Philosophies of Writing Programs
 - ▶ **Quick and Dirty** — get the program working and move on to next project
 - ▶ **Software Engineering** — the discipline of writing programs so they can be understood and maintained by others
- ▶ Programming is an art and skill – **learned by practice**, not rote memorization, much like playing the piano

25

Java & the Object Oriented Paradigm

- ▶ **Paradigm**: an existing theoretical framework or set of rules
- ▶ **Paradigm Shift**: a new idea/framework replaces an older one
- ▶ Old programming paradigm: **procedural** — programs were a series of statements, procedures and functions which operated on openly available data
- ▶ New programming paradigm: **object oriented** — data and operations are grouped together into integrated units called **objects**, providing some security for data integrity
- ▶ Each **object** is an instance of a particular **class**; a single class can serve as a pattern for many different objects.

26

Why Java?

- ▶ **Simple**, efficient object oriented language
- ▶ Capacity to access and expand **libraries** of code
- ▶ **Robust and Secure**: Designed for creating highly reliable software, with security features designed into the language and run-time system
- ▶ **Architecture Neutral and Portable**: Java was designed to work well over a network, regardless of machine type or operating system (multi-platform)

27

Why Java?

- ▶ **High Performance**: runs fast, responds quickly, cleans up after itself
- ▶ **Interpreted, Threaded, and Dynamic**: faster program development, multiple activities at the same time, and constantly evolving
- ▶ **Cost-effective**: open-source freeware is available on the Internet

28

Software Development Report

Internal Documentation: comments within a program, written for other programmers. Required in all programs this semester.

External Documentation: report written for users and other programmers concerning the program — its design, implementation, testing, correctness, and other information

You may be required to develop external documentation for some programs this semester.

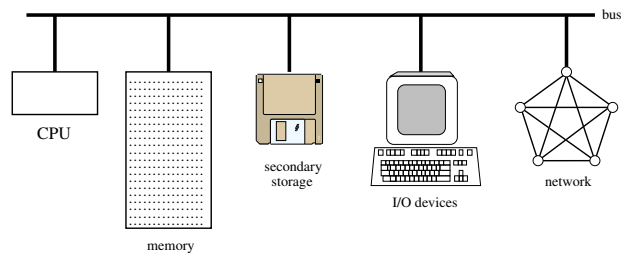
29

What is Computer Science?

- ▶ **Hardware** - tangibles
- ▶ **Software** - abstract, intangible
- ▶ **Problem Solving** - a skill one needs to practice

30

Components of a Typical Computer



31

Computer Hardware

- ▶ **CPU (Central Processing Unit)** - an integrated circuit on a silicon chip; computations, coordinates computer activities
- ▶ **Memory (Primary Storage)** - usually a special integrated-circuit chip called a **RAM**, or *random-access memory*; information lost when machine turned off
- ▶ **Secondary Storage** - hard disk, thumb drive, CDs, diskettes, etc.; permanent data storage
- ▶ **Input/Output Devices (I/O devices)** - keyboard, mouse, monitor, printer
- ▶ **Network** - connection to other computers, **Internet**

32