

COMP 202

Introduction to Computing 1

WELCOME

Focus of the Course

- Introduction to programming using Java
- Aimed at students with LITTLE or NO background in programming and knowledge of Computer Science

Course Outline

- Week 1-2: Introduction
- Weeks 2-3: Java in 2 weeks
 - Variables and assignments
 - if-then-else and loops
 - the class concept
- Week 4: Data and Expression and classes in detail
- Weeks 5-6: Control structures
- Week 7-8: Object-oriented programming
- Week 9: Arrays
- Week 10-13: Advanced Topics

Week 1

- Administrative
- What is programming?
- How does a computer work?
- Our first programs

Instructor Coordinates

- Clark Verbrugge
 - `clump@cs.mcgill.ca`
- McConnell Eng. Building, Room 230
 - `www.sable.mcgill.ca/~clump`
- Office Hours:
 - Wednesday 10:00 am -11:00 am
 - Friday 10:00 am – 11:00am
- Course website:
 - `www.cs.mcgill.ca/~cs202`

Structure

- Placement Quiz
- 3 hours lecture per week
- 9 tutorials
- Lab TA-Office hours
 - 7 hours per week
 - Times will be announced next week
 - TA's will patrol and be in a nearby room

Tutorials: TBA

- Tutorial 1 is next week
 - Several sessions
 - Date, place TBA
- Tutorial 1:
 - SOCS accounts, basics of Unix
 - WebCT
 - DrJava
- Tutorial 2 and 3:
 - basics of programming
- Tutorials 4, 6, 7, 8:
 - advanced programming
- Tutorials 5 and 9:
 - preparation for midterm, exam

What you have to do

- 5 Assignments
 - Part 1
 - Not graded
 - Simple questions and small exercises to reinforce what was learned in class and help study
 - TAs will help during lab hours to solve the problems
 - Part 2
 - Graded
 - Programming tasks
- Midterm + Final
- Labs: Optional, but highly recommended

Grading Scheme

- Assignments: 30%
 - Assignments 1 through 4 have the same weight (5%)
 - Assignments 5: 10%
 - Late penalty
- Midterm: 20%
 - Date to be set soon
- Final: 50%

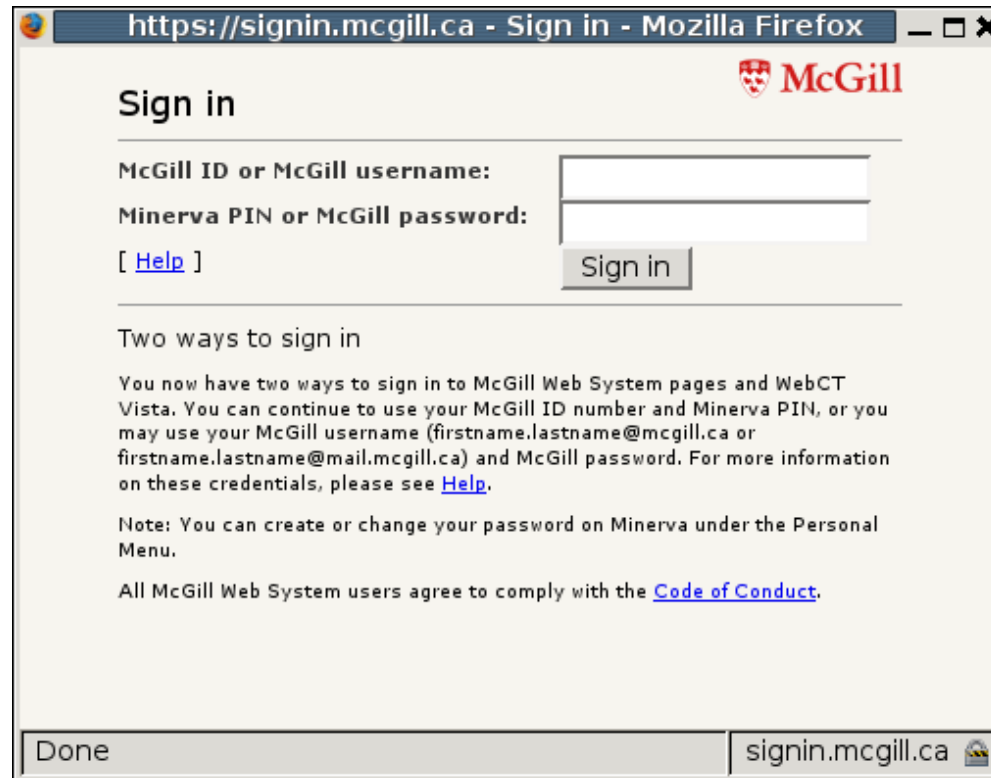
Communication

- WebCT Vista
 - Lecture notes
 - Assignments
 - Frequently asked questions
 - Discussions
 - Announcements
 - ...
- Notes, assigns also on website:
 - `www.cs.mcgill.ca/~cs202`

www.mcgill.ca/webct

The screenshot shows the myCourses (WebCT Vista) website in a Mozilla Firefox browser window. The address bar displays <http://www.mcgill.ca/mycourses/>. The page features a navigation menu with links such as HOME, PROSPECTIVE STUDENTS, STUDENT INFORMATION, ALUMNI & FRIENDS, FACULTY & STAFF, PUBLIC & MEDIA, RESEARCH & INNOVATION, ADMIN & GOVERNANCE, LIBRARY & COLLECTIONS, and FACULTIES & SCHOOLS. A search bar is located on the right side of the navigation menu. The main content area includes a section for myCourses with links to General information, Instructors, Students, and Sign-in help. There is also a section for Announcements with links to myCourses service interruption tonight, VPN access interruptions, and USNET News Service update. A section for Go to myCourses provides instructions on how to access WebCT courses. The Features section includes a link to fyi.mcgill.ca. The Instructors section has links to Getting started, Documentation, Workshops, Consultations, Software, FAQ, and Help. The Students section has links to Getting started, Online tutorials, Software, FAQ, and Help. The footer contains contact information for myCourses (WebCT Vista) and copyright information for McGill University.

Login Screen



The screenshot shows a web browser window titled "https://signin.mcgill.ca - Sign in - Mozilla Firefox". The page has a light beige background and the McGill logo in the top right corner. The main heading is "Sign in". Below it, there are two input fields: "McGill ID or McGill username:" and "Minerva PIN or McGill password:". To the left of the password field is a blue link "[Help]". A "Sign in" button is located to the right of the password field. Below the input fields, there is a section titled "Two ways to sign in" followed by a paragraph explaining the two methods: using a McGill ID number and Minerva PIN, or using a McGill username (firstname.lastname@mcgill.ca or firstname.lastname@mail.mcgill.ca) and McGill password. A "Note" states that users can create or change their password on Minerva under the Personal Menu. At the bottom, it says "All McGill Web System users agree to comply with the [Code of Conduct](#)." The browser's status bar at the bottom shows "Done" on the left and "signin.mcgill.ca" on the right.

https://signin.mcgill.ca - Sign in - Mozilla Firefox

Sign in

McGill ID or McGill username:

Minerva PIN or McGill password:

[[Help](#)]

Two ways to sign in

You now have two ways to sign in to McGill Web System pages and WebCT Vista. You can continue to use your McGill ID number and Minerva PIN, or you may use your McGill username (firstname.lastname@mcgill.ca or firstname.lastname@mail.mcgill.ca) and McGill password. For more information on these credentials, please see [Help](#).

Note: You can create or change your password on Minerva under the Personal Menu.

All McGill Web System users agree to comply with the [Code of Conduct](#).

Done | signin.mcgill.ca

Selection Screen

The screenshot shows a web browser window titled "myCourses (WebCT Vista) - Mozilla Firefox". The address bar displays "https://home.mcgill.ca/mycourses/". The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The toolbar shows navigation buttons and a search bar with "Google". The browser's status bar indicates "myCourses (WebCT Vi..." and "404 Not Found".

The main content area of the web page is titled "myCourses (WebCT Vista)". It features a navigation menu with links: HOME, PROSPECTIVE STUDENTS, STUDENT INFORMATION, ALUMNI & FRIENDS, FACULTY & STAFF, PUBLIC & MEDIA, RESEARCH & INNOVATION, ADMIN & GOVERNANCE, LIBRARY & COLLECTIONS, and FACULTIES & SCHOOLS. Below the menu is a red banner with the McGill logo and the text "myCourses (WebCT Vista)". A search bar with a "Find" button is also present.

The page is divided into several sections:

- myCourses**: A sidebar menu with links to General information, Instructors, Students, and Sign-in help. Below this is a link to "Report a problem / comment".
- Announcements**: A section titled "myCourses service interruption tonight | August 29" and "VPN access interruptions August 30 | August 28". It also includes a link to "USENET News Service update | July 19" and a link to "All Announcements [RSS]".
- Go to myCourses**: A section with links to "WebCT Vista (Summer 2007 ONLY)" and "myCourses (WebCT Vista)".
- Features**: A section titled "For your Information" with a link to "fyi.mcgill.ca".
- Instructors**: A section with links to "Getting started", "Documentation", "Workshops", "Consultations", "Software", "FAQ", and "Help".
- Students**: A section with links to "Getting started", "Online tutorials", "Software", "FAQ", and "Help".

At the bottom of the page, there is a footer with the following information:

- myCourses (WebCT Vista) [Unit detail]**: 688 Sherbrooke Street West [Map], Montreal, Quebec H3A 3R1, [Email]
- Copyright © 2007 McGill University**
- Page last updated:** Jul. 30, 2007 at 5:36 PM

Course List

Blackboard Learning System - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://mycourses.mcgill.ca/webct/urw/lc5122011.tp0/cobaltMainFrame.dowebct?jSESSIONID=0LDhGVNJdON3D

Local McGill Linux Games Stuff Commerce Eclipse Research Diversions Temp Weather Google Google Scholar Dictionary.com

Blackboard Learning S... 404 Not Found 404 Not Found

Firefox prevented this site from opening a popup window. Preferences

My Settings | Check Browser | Help | Log out

McGill myCourses

myCourses Home Content Manager

Welcome, Clark Verbrugge. Today is August 29, 2007 11:35 EDT.

Calendar Day
You currently have no entries for today.

My Grades
You currently have no new grades.

Campus Announcements
[Service Interruption - Wed Aug 29 5PM - Aug 30 9AM](#)

Course List

[Concurrent Programming - Fall 2007 - COMP-409-001](#)
Section Instructor: Clark Verbrugge
My Role: Section Designer/Section Instructor

[Concurrent Programming Lang - Fall 2007 - COMP-623-001](#)
Section Instructor: Clark Verbrugge
My Role: Section Designer/Section Instructor

[Intro to Computing 1 - Fall 2007 - COMP-202-002](#)
Section Instructor: Clark Verbrugge, Ladan Mahabadi, Mathieu Petitpas
My Role: Section Designer/Section Instructor

Calendar Week
You currently have no entries for this week.

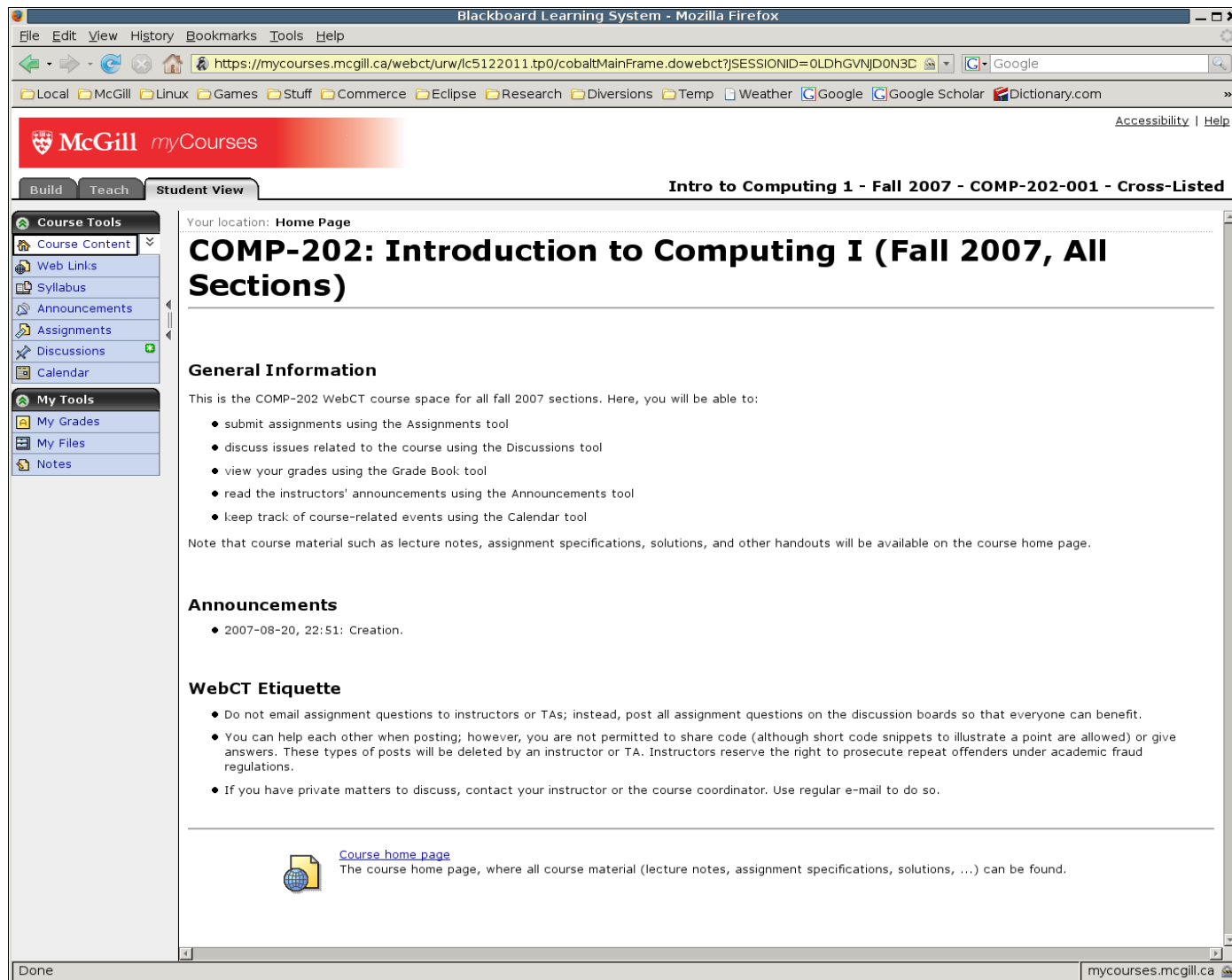
Campus Bookmarks
You currently have no campus bookmarks.

Personal Bookmarks
You currently have no bookmarks.

Who's Online
[Concurrent Programming - Fall 2007 - COMP-409-001](#) (1)
[Concurrent Programming Lang - Fall 2007 - COMP-623-001](#) (1)
[Intro to Computing 1 - Fall 2007 - COMP-202-002](#) (3)

Done mycourses.mcgill.ca

The Course



The screenshot shows a web browser window titled "Blackboard Learning System - Mozilla Firefox". The address bar displays the URL: <https://mycourses.mcgill.ca/webct/urw/c5122011.tp0/cobaltMainFrame.dowebct?SESSIONID=0LDhGVNjDON3D>. The browser's bookmark bar includes links to Local, McGill, Linux, Games, Stuff, Commerce, Eclipse, Research, Diversions, Temp, Weather, Google, Google Scholar, and Dictionary.com.

The Blackboard interface features a red header with the McGill logo and "myCourses" text. Below this, a navigation bar includes tabs for "Build", "Teach", and "Student View". The course title "Intro to Computing 1 - Fall 2007 - COMP-202-001 - Cross-Listed" is displayed on the right.

The left sidebar contains two main sections: "Course Tools" and "My Tools". "Course Tools" includes links to Course Content, Web Links, Syllabus, Announcements, Assignments, Discussions, and Calendar. "My Tools" includes links to My Grades, My Files, and Notes.

The main content area is titled "Your location: Home Page" and displays the course title "COMP-202: Introduction to Computing I (Fall 2007, All Sections)". Below the title, there is a section for "General Information" which states: "This is the COMP-202 WebCT course space for all fall 2007 sections. Here, you will be able to:" followed by a bulleted list:

- submit assignments using the Assignments tool
- discuss issues related to the course using the Discussions tool
- view your grades using the Grade Book tool
- read the instructors' announcements using the Announcements tool
- keep track of course-related events using the Calendar tool

A note follows: "Note that course material such as lecture notes, assignment specifications, solutions, and other handouts will be available on the course home page."

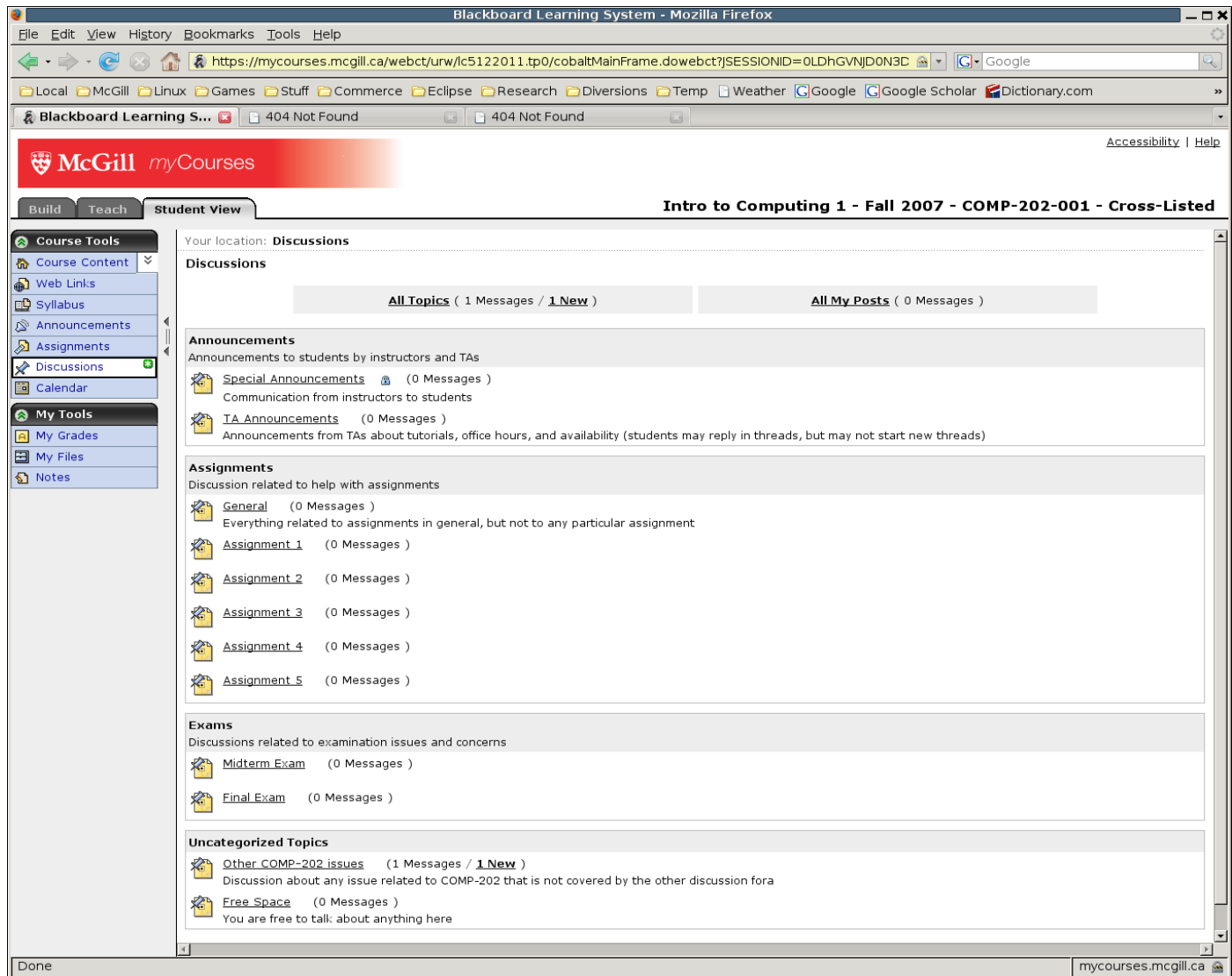
The "Announcements" section shows a single entry: "2007-08-20, 22:51: Creation."

The "WebCT Etiquette" section provides guidelines:

- Do not email assignment questions to instructors or TAs; instead, post all assignment questions on the discussion boards so that everyone can benefit.
- You can help each other when posting; however, you are not permitted to share code (although short code snippets to illustrate a point are allowed) or give answers. These types of posts will be deleted by an instructor or TA. Instructors reserve the right to prosecute repeat offenders under academic fraud regulations.
- If you have private matters to discuss, contact your instructor or the course coordinator. Use regular e-mail to do so.

At the bottom, there is a link to the "Course home page" with a document icon, accompanied by the text: "The course home page, where all course material (lecture notes, assignment specifications, solutions, ...) can be found."


Discussions



The screenshot shows a web browser window titled "Blackboard Learning System - Mozilla Firefox". The address bar displays the URL: <https://mycourses.mcgill.ca/webct/urw/lc5122011.tp0/cobaltMainFrame.dowebct?SESSIONID=0LDhGVNjD0N3D>. The browser's bookmark bar includes links to Local, McGill, Linux, Games, Stuff, Commerce, Eclipse, Research, Diversions, Temp, Weather, Google, Google Scholar, and Dictionary.com. The page header features the McGill myCourses logo and navigation tabs for Build, Teach, and Student View. The course title "Intro to Computing 1 - Fall 2007 - COMP-202-001 - Cross-Listed" is displayed on the right. The left sidebar contains a "Course Tools" menu with links to Course Content, Web Links, Syllabus, Announcements, Assignments, Discussions (highlighted), Calendar, My Tools, My Grades, My Files, and Notes. The main content area, titled "Your location: Discussions", shows a summary of discussion topics. At the top, there are two tabs: "All Topics (1 Messages / 1 New)" and "All My Posts (0 Messages)". The content is organized into several sections: "Announcements" (with sub-sections for Special Announcements and TA Announcements), "Assignments" (with sub-sections for General and five numbered assignments), "Exams" (with sub-sections for Midterm Exam and Final Exam), and "Uncategorized Topics" (with sub-sections for Other COMP-202 issues and Free Space). The status bar at the bottom indicates "Done" and the URL "mycourses.mcgill.ca".

Assignments

Assignments

 **Assignments**

Inbox [Submitted](#) [Graded](#) [Published](#)

Assignments that you have been assigned and submissions that have been returned to you for editing.

There are currently no Assignments in your Inbox.

What is programming?

What do other sciences do?

- Life Science:
 - Example: study how a cell works
 - cell is something complex that already exists
 - researcher tries to figure out how this complex cell works
 - get background knowledge
 - learn laboratory skills
 - make hypothesis
 - run many experiments to see whether hypothesis works
- Engineer
 - Example: build a bridge
 - Create something new
 - Must follow many rules --> must work
 - Must know the math, materials, physics,...
 - What is more complex: a cell or a bridge?

What do Computer Scientists do?

- Example: build the web-portal for an online bookstore
- Create something new
 - Webpages
 - Functionality:
 - browse with keywords
 - Maintain your shopping card
 - Perform checkout
- What is more complex: a cell or a piece of software?

Let's start small:

Recipe for Scrambled Eggs

- Ingredients: two eggs, tablespoon of oil, salt
- Instructions:
 - Add oil to pan
 - Heat pan on stove
 - Crack eggs into pan
 - Add salt
 - Mix until light and flakey
- Output: scrambled eggs

What if we
did not follow
the order?

A recipe is a
series of steps

Currency Exchange

1. Input:

- *Amount*
- *Source Currency*
- *Desired Currency*

• Instructions

- Look up in table current *exchange rate* for the selected currencies
- Calculate *result* as $\text{amount} * \text{exchange rate}$

• Output: *result*

What is a program?

- A program solves a particular task
- A program is a list of instructions
- The list must be ordered correctly
- A program has inputs and outputs
- Each instruction tells the computer to do something (an action, a calculation, a comparison)

What is software?

- Software = Program ?
- Software typically consists of many subcomponents / modules / subprograms
- Each subprogram solves a particular task
- Software also includes the data to be used and manipulated
 - input / output data
 - internal data (e.g, table with exchange rates)

The bank machine: functionality

- Withdraw money from checking account
- Show balance on checking account
- Show balance on saving account
- Transfer money from checking account to saving account
- Transfer money from saving account to checking account
- Deposit money on checking account
- Deposit money on saving account
- Make bill payment from checking account
- ...

The bank machine: Interaction

- Login (sub-program)
- Task menu (each being a sub-program)
 - Withdraw
 - Balance
 - Transfer
 - Deposit
 - Bill payment
- Once task is selected walk-through for specific task on hand
 - requires different input data, produces different output
- Allow many tasks in one session
- logout

The bank machine: login

- Input: pin
- If pin is correct
 - Result: go to task menu
- Else get new pin as input
- If pin is correct
 - Result: go to task menu
- Else get new pin as input
- If pin is correct
 - Result: go to task menu
- Else print warning, do not release card

Bank machine: money transfer

- Input ?
- Instructions?
- Output?

Goal of this Course

- Translate a complex task into the software that executes this tasks
 - *Structure* the problem space into many small sub-tasks
 - Write programs for each subtask
 - *Connect the programs* to build software
- At the end of course, you will be able to write a program that simulates a bank machine (and many other things)
 - Of course, since we don't have the machines themselves, no “real” money and “real” cards are involved

The Java Programming language

- A *programming language* specifies the words and symbols that we can use to write a program
- A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid *program statements*
- Java was created by Sun Microsystems, Inc.
- It was introduced in 1995 and has become quite popular
- It is an object-oriented language

Our first Java Program

```
public class MyFirstProgram
{
    public static void main (String [] args)
    {
        // let's say hello
        System.out.println("hello world!");
    }
}
```

Output: hello world

Program Structure (for now)

- Each program is a file
- Each program describes one *class*
- A *class* describes a real-life concept
 - Bank machine, Calculator, System.out, MyFirstProgram
- A *class* has data and methods (functions, operations)
- Example methods
 - *main method* (most classes have one main method)
 - *println*
 - In the example, we provide an implementation for *main* and we use *println* (somebody else has already programmed this method; we simply call it)
- A method contains statements
 - Statements are the instruction steps that are executed when the program runs
- Ignore *public*, *static*, *void*, *args* for now

Java Program Structure

```
public class MyProgram
```

```
{
```

```
}
```

class body

class header: The name of the class

Important: The class header name **MUST** be the same name as the file name:
MyProgram.java


Comments can be added almost anywhere

Java Program Structure

```
public class MyProgram
{

    public static void main (String[] args)
    {
    }

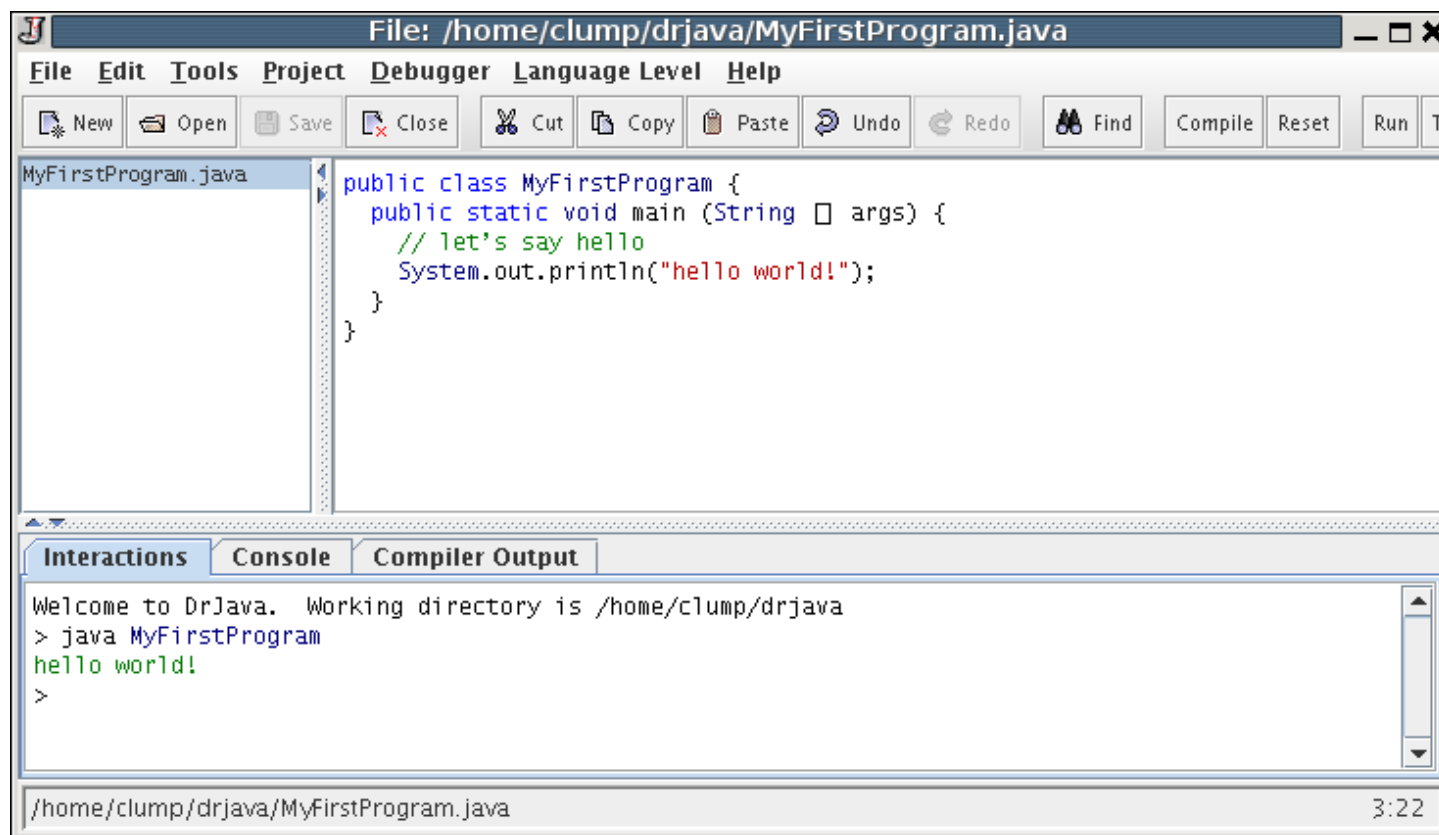
}
```



Syntax is important

- Natural Languages are ambiguous
- Programming languages may not!
- Examples of Java syntax rules
 - The class name is the same as the file name
 - Parenthesis
- Comments
 - Describe purpose and programming steps
 - Do not affect what the program does
 - Syntax:
 - `// this is a comment`
 - `/* this is a comment */`

DrJava



Compiling and running a program in DrJava

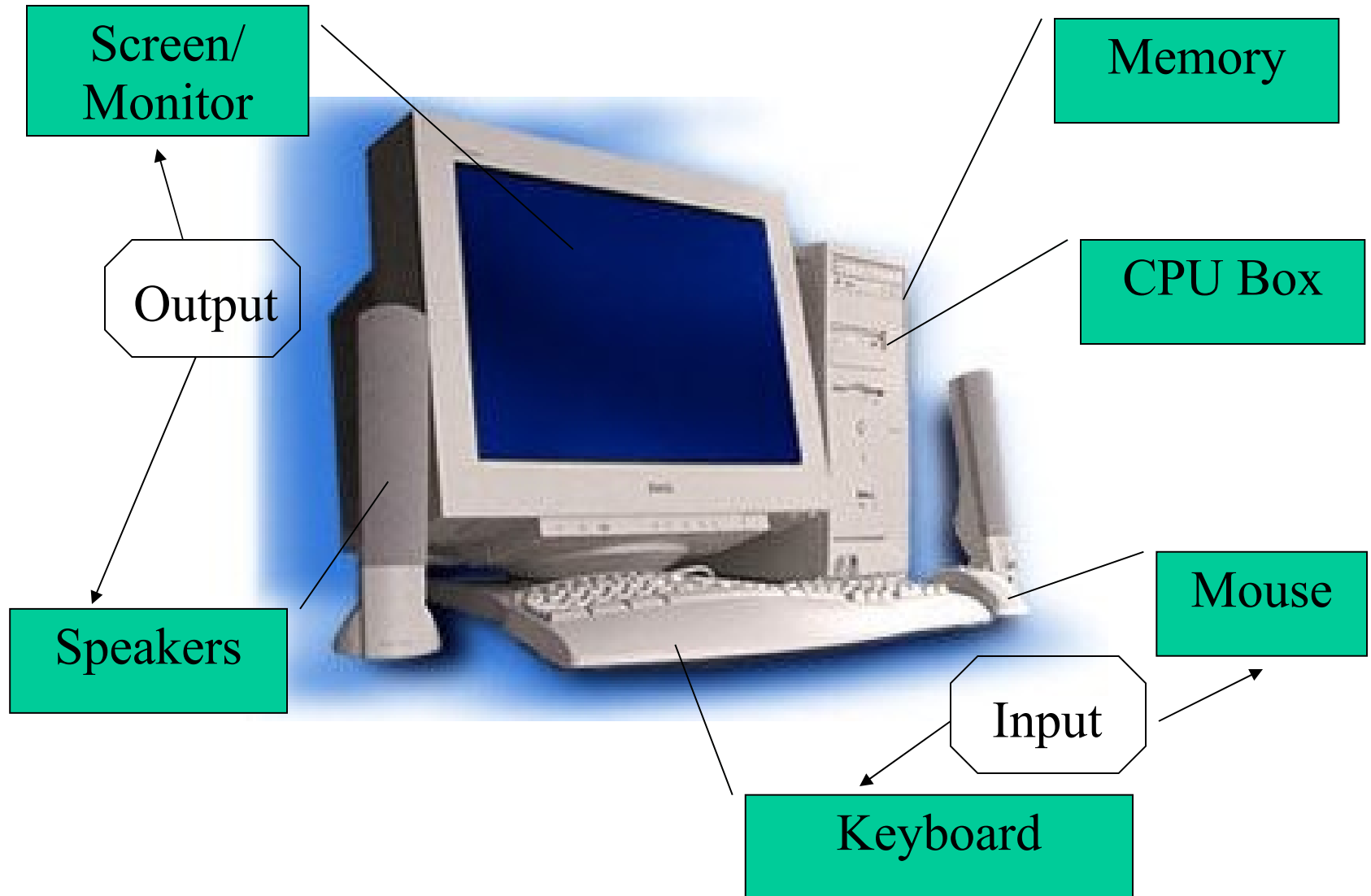
- Type in your program
- Save the program
 - Store all your files in one directory for now
 - Give the program the same name as the class
- Compile the program
 - this produces a .class file
 - Translates the program into something the computer can understand and execute
- Run the program
- Observe the result and adjust the program if necessary

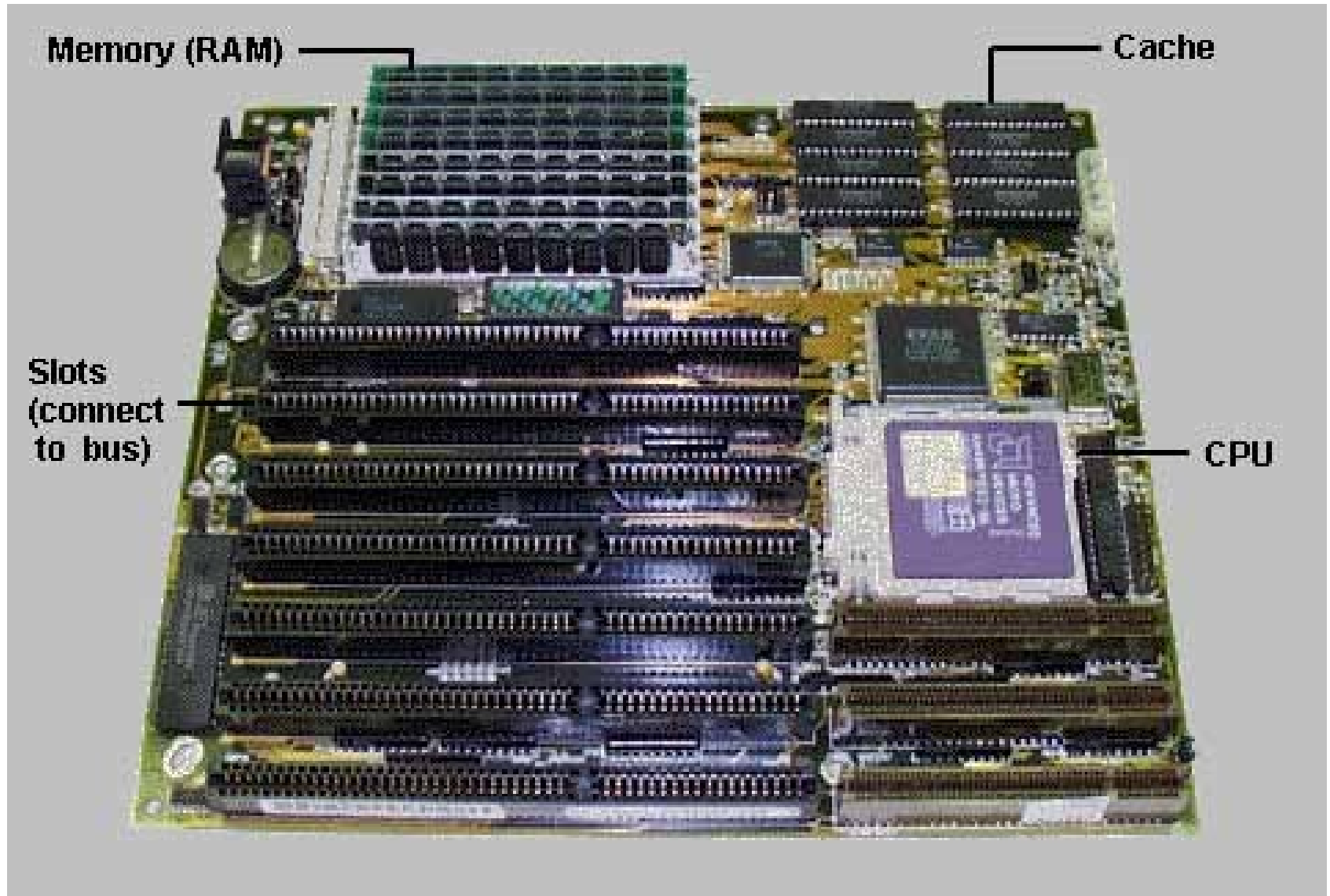
How a computer works

Hardware and Software

- Hardware
 - the physical, tangible parts of a computer
 - keyboard, monitor, wires, chips, mouse
- Software
- A computer requires both hardware and software
- Each is essentially useless without the other

The Personal Computer





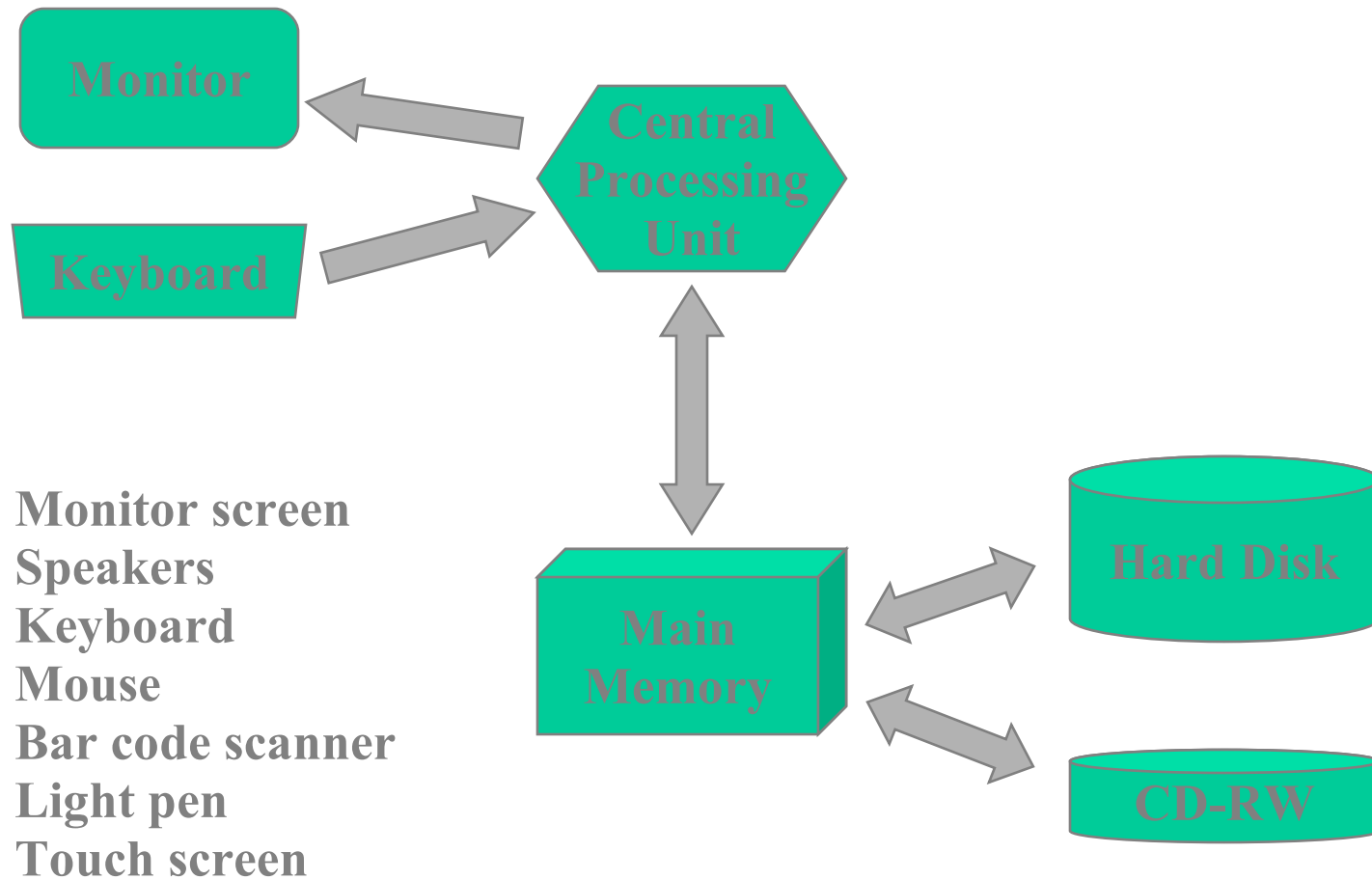
CPU (Central Processing Unit)

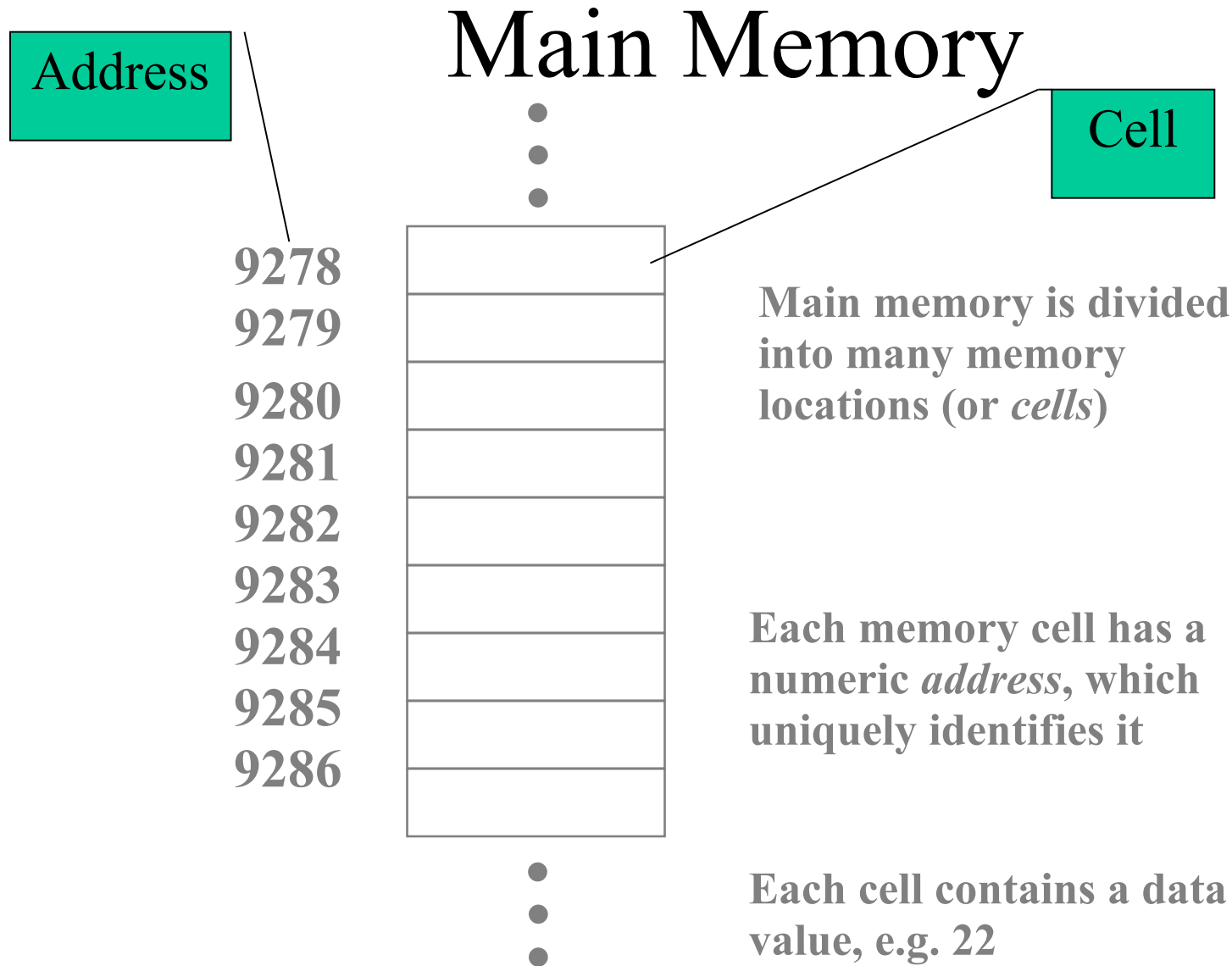
- Performs instructions
 - Arithmetic operations
 - Much more fine-grained than what we have seen so far
 - Controls the information / data in a computer

Memory

- Memory holds the data
- Main memory (most of it called RAM):
 - Data has to be in main memory so that CPU can access it
 - Volatile: lost when program exits; computer shuts off
- Disk / CD
 - Persistent
 - This is where you keep the data for long-term storage

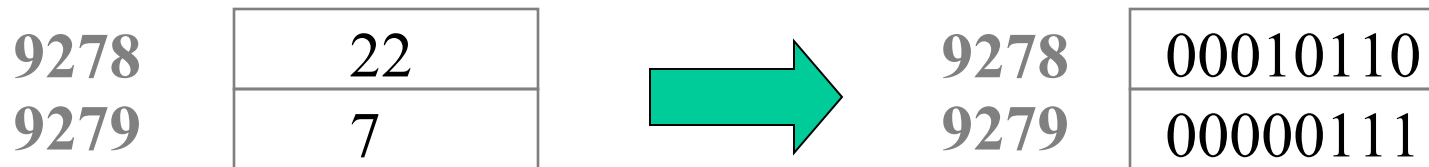
Interaction





Bits and Bytes

- Bit -- most basic unit of memory
 - 1 or 0, on or off
- 1 Byte = 8 bits
- In a computer, data values are stored as a sequence of bits

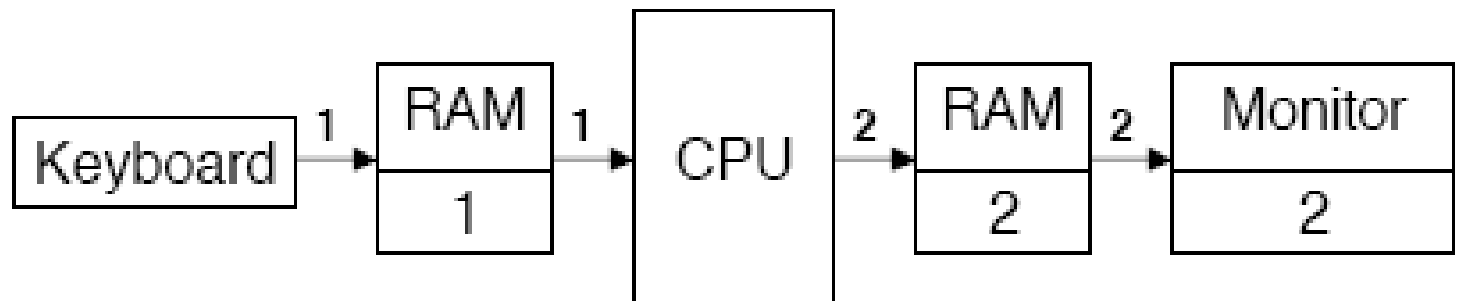


Program execution

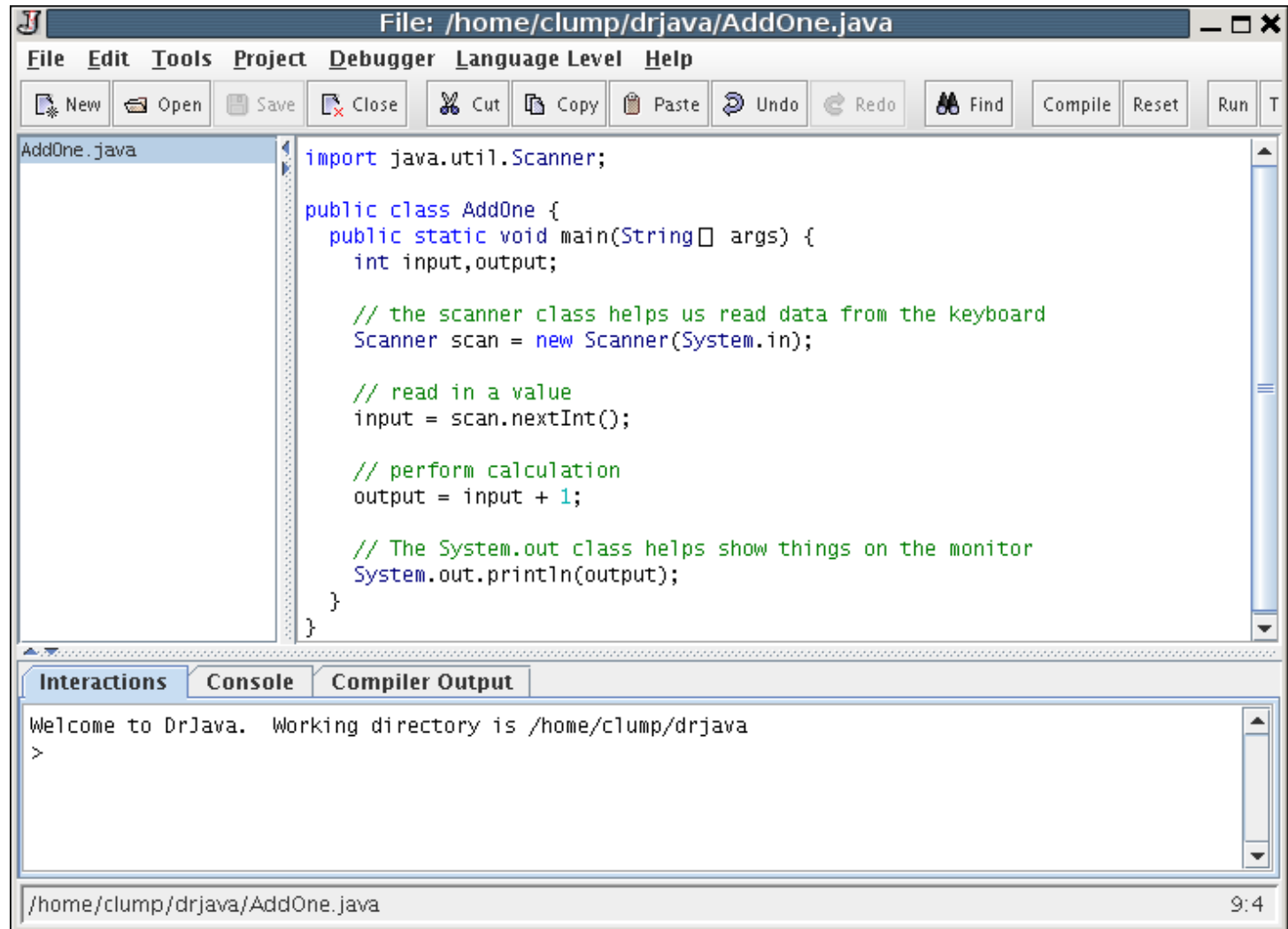
- A program tells the CPU how to manipulate and/or move information
- Programming is about processing information
 - Take some input, manipulate it in some way, and produce a particular output

Program / CPU / Memory

- Example:
 - Input read a number from keyboard
 - Add 1 to it
 - Output it on screen



In Java



Improving our understanding

- Compile the program
 - The CPU does not understand the Java language!
 - Java Language is translated into language that the CPU understands
- To build this “program” we'll need 3 classes:
 - Each class fulfills a task
 - System.out: allows us to print things on the monitor
 - Scanner: allows us to read input from keyboard
 - AddOne: that is the class that implements our task!

Variables

- A variable is a placeholder for a cell/location in main memory (or a consecutive block of cells)
- Each variable has a type
 - For now: integers (int)
- Each variable has a name
 - This is how we can refer to the data
 - Much easier than using an address
- Each variable has a value
 - The content of the cell (or set of cells) the variable is referring to
 - An int has 4 Bytes, thus it uses 4 consecutive cells

input (int)



Variable Declaration

- Variable Declaration
 - Typically at begin of method
 - Must indicate type, name
 - `int input;`
 - Can indicate several variables in one statement
 - `int input, output;`
 - Can additionally indicate value
 - `int input = 1;`

Using a Variable

- Variable Assignment Statement
 - Throughout the program we can assign values to a variable
 - `input = scan.nextInt();`
 - `output = input + 1;`
- The assignment operator is the = sign
- The *expression* on the right is evaluated and the result is stored in the variable on the left
 - The old value of the left variable is overwritten
- Reading
 - We can read the value
 - `output = input + 1;`
 - `System.out.println(output);`

println so far

- *println* takes one input
 - A character string: *println("hello world")*
 - The value of a variable: *println(output);*
 - This value is automatically converted into a character string that can be printed

What we learned so far

- What is a program?
- How does a simple Java program look like?
- How is a simple Java program executed on a computer?
- Fundamentals:
 - A program has operations/statements/instructions
 - They are executed by the CPU
 - A program uses data
 - Data is stored in main memory
 - Data is accessed via the variable concept
 - Different tasks are implemented via different classes