CS 211 Recitation 1

Section 6

Meghan Desai

- Senior, ECE
- mpd25@cornell.edu
- Office Hours:
  - Monday 1:00pm – 2:00pm
  - Upson 328
- Consultant for 2 semesters
- TA for 2 semesters
- Took CS211 in FALL 2002

Greg Studer

- __________, CS
- gms25@cornell.edu
- Office Hours:
  - ______________
  - Upson 328
- Consultant for ___________
- TA for _________
- Took CS211 in __________

CS211.TA.Admin

- Sign the sheet of records
  - WHY?
  - For emergency contact, room change, etc.
- What's the point of a section?
  - Supplement Lecture
  - Cover material not covered in Lecture
  - Additional Problems/Exercises…
  - Question & Answer

CS211.course_questions

- Any questions about the course?

The `main()` method

- Application
  - Runs on its own
  - Executes *only* the commands in the main method
    - By function calls, executes rest of code…
- **`public static void main(String[] args)`**
  - `public` – allows *outside* source to access method
  - `static` – will be discussed later in course
  - `void` – method returns nothing
  - `args` – will be discussed later this section

The COMMAND Line

- Any terminal that excepts typed commands:
  - Example: DOS, UNIX, etc.
- Download and install Sun's Java 5 JDK
  - Right Click on 'My Computers' → 'Properties'
  - Click 'Advanced' Tab
  - Click 'Environment Variables'
  - Find system variable 'PATH' and click edit
  - At the end, add ; and the location of your java bin dirctory
- Windows:
  - Start→Run→CMD→ENTER
  - DOS window pops up…
  - **`C:\>_`**

Using the COMMAND Line

- Navigate to directory with `.java` files…
  - **`C:\>dir mydir`**
  - **`C:\mydir\>_`**
- Compile your java program
  - `main()` method in class `myClass.java`:
  - **`C:\mydir\>javac myClass.java`**
  - If more than one class:
  - **`C:\mydir\>javac *.java`**
- This produces `.class` files

Running `myClass`

- Run your java program:
  - `C:\mydir\>java myClass`
- Automatically finds filename.class file…
- Can only name one class, no *
- Runs the program on JVM
  - Java Virtual Machine

COMMAND Line Arguments

- Can pass in arguments to a java program:
- When running:
  - `C:\mydir\>java myClass arg0 arg1 …`
- They appear in the **`String[] args`** array
  - args[0]="arg0"
  - args[1]="arg1"

Indexing starts at 0

- What if you wanted to send in integers?

DEMO

```
public class myClass{
  public static void main(String[] args)
  {   int a = Integer.parseInt(args[0]);
      System.out.println(a);
  }
}
```

DEMO

QUESTIONS?