CMPU125
Topics in Computer Science

Lecture 1

Jenny Walter

Fall 2008

Goals of this Course

- Introduce concepts and techniques of computer science.
- Course work will include writing and analyzing computer programs.
- Java, an object-oriented programming language, will be our tool.

Prerequisites

- Some previous experience in programming and/or computer science.
- Otherwise, talk to Prof. Walter to see if this is the right course for you.
- If you have sufficient experience, it may be better for you to take CMPU102 which meets MWF at 1:30

Topics to be Covered

- A review of CS1 (CMPU101) and all of CS2 (CMPU102) in one semester.
- I will review Java programming, starting at the very beginning.
- But I will go through it quite quickly.
- Take a look at the topics on the web page and on the syllabus.

Course Web Page

https://www.cs.vassar.edu/~cs125

- Overview of the course.
- Schedule of topics and readings.
- Lab and homework assignments.
- Summary of grading policy.
- Professor Walter's lecture notes.

Communication

- I often send email messages to the entire class.
- My messages typically contain hints or clarifications of course material.
- I will send email messages to individuals if I have specific comments or questions for that person

Contacting Professor Walter

- Office: 124 OLB
- Phone: 437-7449
- Email: walter@cs.vassar.edu or jewalter@vassar.edu

Office Hours

- Mon-Thu: 2pm-4:30pm
- Also by appointment

Reading

- Carrano and Prichard, "Data Abstraction and Problem Solving with Java" (Second Edition), Addison-Wesley 2006. (Required)
- The Java Tutorial, Sun Microsystems, Available on line at: http://java.sun.com/docs/books/tutorial/ (Recommended resource.)

Labs

- One lab each week.
- Friday, 3:10pm-4:40pm.
- OLB 104.
- Required.

Homework

- One programming assignment each week.
- Absolutely essential for learning the course material.
- Try not to fall behind, since it can be difficult to catch up.

Grading

- Labs: 10%.
- Assignments: 25%
- Midterm Exam 1: 20%
- Midterm Exam 2: 20%
- Final Exam: 20%
- Class Participation: 5%

Deadlines and Lateness

- Assignments will generally be due on Thursdays by midnight. Any exceptions will be announced.
- Late assignments will be accepted with a penalty and will not be accepted once solutions have been distributed.

Coaching

- There will be a coaches (teaching assistants) assigned to our class.
- The coaches will answer questions in the lab and will hold office hours in the evenings and on weekends.
- Keep an eye on the web page for coaching schedule.

Examinations

- Open book and open notes.
- Cumulative, by the nature of the material.
- Exam dates shown on the course web page and in the syllabus.

Academic Integrity

- You may discuss general ideas with classmates.
- You must do each programming assignment entirely by yourself.
- You may not discuss or share programs with other students.
- Vassar regulations require the professor to report suspected violations of academic integrity to the Dean of Studies.
- Read the “Originality and Attribution” pamphlet.

Disability Services

- Academic accommodations are available for students with disabilities but you must be registered with the Office of Disability and Support Services.
- If you need disability accommodations, please schedule an appointment with me early in the semester and bring your DSS letter.

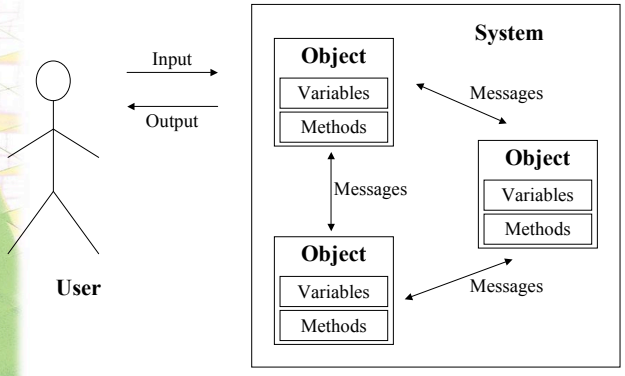
END OF COURSE INTRODUCTION…

QUESTIONS?

Java

- Why was Java invented?
  - To make programs portable via the internet.
- Why do we teach Java?
  - It's a widely-used programming language.
  - Has a minimum of troublesome details.
  - Learning a high-level programming language allows you to learn other programming languages faster.

Object Oriented Programming



Program Structure

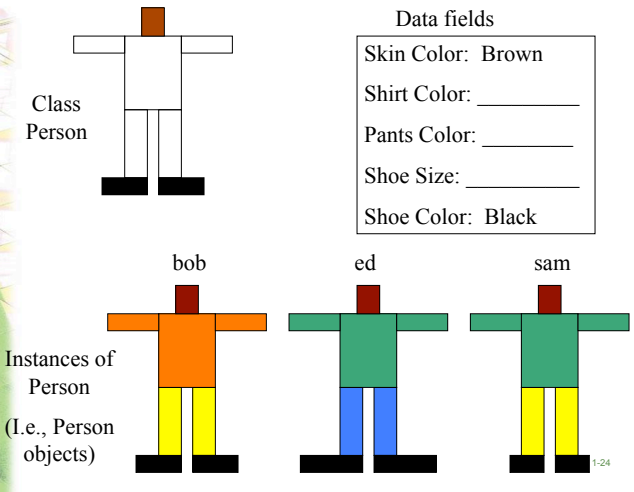
- Java program basic elements:
  - Packages
  - Classes
- Java Class elements:
  - Data fields
  - Methods

Java Programming

- Java programming involves mix of
  1. using standard, predefined classes documented in the Java API and
  2. writing and documenting your own classes.
- Because of your previous experience writing Java code, you have already used Java library classes including
  - String, System, Integer, Double, Math
- This semester, you should become *much* more familiar with using the Java API = Application Programming Interface.

Classes and Instances

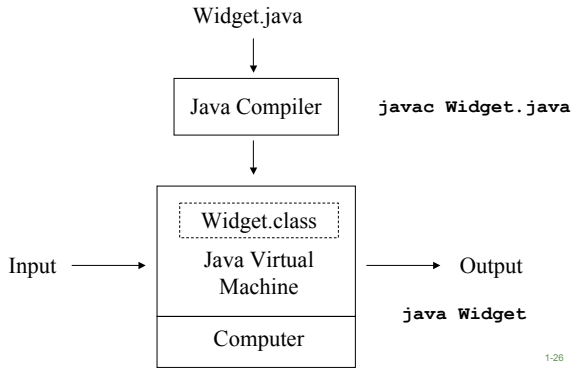
- A *class* defines a data type.
  - Specifies properties of all objects in the class.
  - Indicates ways in which objects in the class may differ from each other.
  - Defines ways in which objects in the class may communicate with each other and with the user.
- An *instance* is an individual object.
  - Sharing the common properties of the class.
  - Manifesting properties that distinguish it from other objects in the class.



Java Development Process

- Programmer types Java source code.
- Programmer stores source code in file “Widget.java”.
- Compiler translates source code into Java byte code.
- Compiler stores byte code in file “Widget.class”.
- Compiled code runs on the Java Virtual Machine.

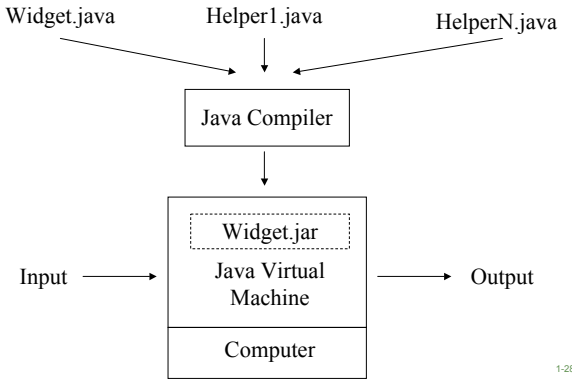
Java Development Process



Java Development Process

- Programmer prepares Java source code.
- Programmer stores source code in files:
  `Widget.java, Helper1.java, … , HelperN.java`
  that are part of the same package.
- Compiler translates source code into Java byte code.
- Package compressed into file `Widget.jar`
- Compiled code runs on the Java Virtual Machine.

Java Development Process



NetBeans

- Integrated Java Development Environment.
- Editor, Compiler, Debugger, and more.
- Free version available for Windows, Mac and Linux.
- See link to download site on our class web page.
- Be sure to download NetBeans bundled with JDK5.0 (third column from the left with "Java SE" for heading - 31 MB).

acm package

We used the acm package in CMPU 101 to

1. Simplify the creation of graphics programs (GraphicsProgram)
2. Simplify read/write operations (ConsoleProgram)

The acm package also has its own API documentation. Links to download the acm.jar file and to the acm API are on the course web page.

Projects

- A single program (called a *project* in most IDEs) is usually composed of a set of one or more interacting classes.

- One of these classes must contain the line of code where execution will begin: the *main method*

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

(IDE = Integrated Development Environment)