Lecture1: Introduction

Bohyung Han
CSE, POSTECH
bhhan@postech.ac.kr
Administration

• CSED233: Data Structures
  ▪ Instructor: Prof. Gary Geunbae Lee (gblee@postech.ac.kr, Eng2-221)
  ▪ TA: TBD
  ▪ Time & Location: TTh 9:30-10:45, TBD
  ▪ Office hour
    • Gary Geunbae Lee: Tue 11:00~12:00 or by appointment
    • TA:
  ▪ Prerequisite: C or C++
  ▪ Course webpage: http://isoft.postech.ac.kr/Course/Data_Structure/Data_Structure_2015S.htm
Textbooks

• For reference
  ▪ Data Structures and Algorithms in Java (2nd Edition) by R. LaFore, Sams Publishing
(Tentative) Schedule

• Introduction to data structures and Java (1 week)
• Algorithm analysis (1 week)
• Arrays and linked lists (2 weeks)
• Stacks and queues (2 weeks)
• Sorting (2 weeks)
• Trees (1.5 weeks)
• Hash (1 week)
• Heap (1.5 weeks)
• Graph algorithms (2 weeks)
Grading

• Assignments (30%)
  ▪ 5~6 assignments
  ▪ Problem solving
  ▪ Programming projects

• Midterm exam (35%)

• Final exam (35%)
  ▪ Comprehensive

• Individual percentages are subject to change.
Assignments

• Programming assignments should be implemented in Java.

• Correctness and efficiency
  ▪ Your assignment will be evaluated based on efficiency as well as correctness.

• Programming projects must be individual efforts.
  ▪ Late assignments will be accepted up to 3 days with penalty. (10% per day)
  ▪ Assignments that have substantial similarities (e.g. copy) will be identified via assignment checking software.

• Submission
  ▪ Through POSTECH LMS (http://lms.postech.ac.kr)
Course Policy

• Attendance
  ▪ Unofficial attendance check
  ▪ I will ask questions in class frequently.

• Academic integrity
  ▪ Make sure to acknowledge the POSTECH academic integrity. Violating the academic integrity means the automatic failure (F) in this class with NO exception.

• All written communication should be in English.
  ▪ Homeworks, reports, exams, etc.
Important Dates (Tentative)

- Assignment Dues

- Midterm exam
  - 4/23 (Thu) in class

- Final exam
  - Comprehensive
  - 6/18 (Thu): Time and location will be announced later.

- Days that need schedule adjustments
  - 3/26, 4/21, 4/23, 6/2, 6/4
Additional Comments on This Course

• This is not an introductory programming language course.
  ▪ I assume that students have basic knowledge about object-orientation and (at least) C/C++.
  ▪ Programming capability is one of the most important grading criteria of this course.

• What to learn in this course?
  ▪ Problem solving skills: theoretical core of computer science
  ▪ Programming languages: JAVA (object-oriented)
Data Structures
What are Data Structures?

• Definition & concept (from Wikipedia)
  ▪ A particular way of storing and organizing data in a computer so that it can be used efficiently.
  ▪ Data structures provide a means to manage large amounts of data efficiently, such as large databases and internet indexing services.
  ▪ Efficient data structures are a key to designing efficient algorithms.

• Examples of data structures
  ▪ Matrices
  ▪ Linked lists
  ▪ Priority queues: stack, queue, etc.
  ▪ Hash tables
  ▪ Trees and graphs
Why Using Data Structures?

- Representing data in an abstract form
  - Storing and organizing data
  - Providing methods to access and use the data systematically

<table>
<thead>
<tr>
<th>Unstructured data</th>
<th>Structured data</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 7</td>
<td>9 4 17 3 6 22 5 7 20</td>
</tr>
<tr>
<td>9 4</td>
<td></td>
</tr>
<tr>
<td>5 17 22</td>
<td></td>
</tr>
<tr>
<td>6 20</td>
<td></td>
</tr>
</tbody>
</table>

Can be defined with array and accessed with index
Why Using Data Structures?

• Implementing algorithms efficiently
  ▪ Data structures are key components to solve problems efficiently.
  ▪ In a large scale problem, the efficiency may be very critical.

• Example: finding a number from a set of numbers
  ▪ How many comparisons do we need to retrieve 7?

8 comparisons

4 comparisons
Programming Languages vs. Data Structures

• Data structures are often implemented with programming languages.
• Any programming language can be used to represent data structures
  ▪ C/C++
  ▪ Java
  ▪ Perl
  ▪ Python
  ▪ Matlab
  ▪ ML
  ▪ etc.
Java Basics
What is Java?

- **History**
  - Inventors: James Gosling, Mike Sheridan, and Patrick Naughton
  - Name: Oak -> Green -> Java
  - Sun Microsystems released Java 1.0 in 1995.

- **Characteristics**
  - General purpose and powerful
  - Object-oriented
  - Platform independent: Write Once, Run Anywhere (WORA)
  - Write less and better code
    - C has 2.0 times longer codes.
    - C++ has 1.4 times longer codes.
Java Usage

TIOBE Programming Community Index for August 2013

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td><strong>↑</strong></td>
<td>Java</td>
<td>15.978%</td>
<td>-0.37%</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td><strong>↓</strong></td>
<td>C</td>
<td>15.974%</td>
<td>-2.96%</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td><strong>↑</strong></td>
<td>C++</td>
<td>9.371%</td>
<td>0.04%</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td><strong>↓</strong></td>
<td>Objective-C</td>
<td>8.082%</td>
<td>-1.46%</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td><strong>↑</strong></td>
<td>PHP</td>
<td>6.694%</td>
<td>+1.17%</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td><strong>↓</strong></td>
<td>C#</td>
<td>6.117%</td>
<td>-0.47%</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td><strong>=</strong></td>
<td>(Visual) Basic</td>
<td>3.873%</td>
<td>-1.46%</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td><strong>=</strong></td>
<td>Python</td>
<td>3.603%</td>
<td>-0.27%</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td><strong>↑↑</strong></td>
<td>JavaScript</td>
<td>2.093%</td>
<td>+0.73%</td>
<td>A</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td><strong>=</strong></td>
<td>Ruby</td>
<td>2.067%</td>
<td>+0.38%</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td><strong>↓↓</strong></td>
<td>Perl</td>
<td>2.041%</td>
<td>-0.23%</td>
<td>A</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td><strong>↑↑↑</strong></td>
<td>Transact-SQL</td>
<td>1.393%</td>
<td>+0.54%</td>
<td>A</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td><strong>↑</strong></td>
<td>Visual Basic .NET</td>
<td>1.320%</td>
<td>+0.44%</td>
<td>A</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td><strong>↓↓</strong></td>
<td>Delphi/Object Pascal</td>
<td>0.918%</td>
<td>-0.09%</td>
<td>A--</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td><strong>↑↑↑↑</strong></td>
<td>MATLAB</td>
<td>0.841%</td>
<td>+0.31%</td>
<td>A--</td>
</tr>
</tbody>
</table>

One of the most popular programming languages in use, particularly for client-server web applications.
Object-Oriented Language

• Everything is an object.
• Every object belongs to a class.
• Class consists of data members, and methods.
  ▪ Data members: the fields of the class
  ▪ Methods:
    • constructors, accessors, mutators, other methods
    • Methods that perform an action
    • Methods that return a value
• Three characteristics
  ▪ Encapsulation: binds processing function to the data (increased modularity)
  ▪ Inheritance: creates classes in hierarchies (knowledge passed down)
  ▪ Polymorphism: allows same function to behave differently
Hello, Java!

All codes in a Java program must belong to a class.

Anyone can run this program.

HelloJava.java
File name should be same with class name.

Class name

public class HelloJava {

public static void main(String[] args) {

System.out.println("Hello, Java!");

}

}

This method belongs to the class, but not an object.

Anyone can run this method.

The method we want to call

The parameters passed to this method (array of strings).
Java Virtual Machine (JVM)

• What is JVM?
  ▪ A process virtual machine that can execute Java byte code
  ▪ Implemented on operating systems or (sometimes) hardwares
  ▪ Distributed along with Java Class Library, a set of standard class libraries that implement the Java API
  ▪ Realize platform independence

• Compiling and running Java codes on command line
  ▪ Compile: $javac$ HelloJava.java
  ▪ Execute bytecode: $java$ HelloJava

Generate bytecode: HelloJava.class
Platform Independence of Java

Stage 1: Compilation

Java program

Filename.java

Java compiler (javac)

Filename.class

Java bytecode (generic binary)

Stage 2: Interpreting and executing the byte code

Filename.class

Java bytecode (generic binary)

Java interpreter (java)

Write Once, Run Anywhere (WORA)

Machine language instruction (UNIX)

Machine language instruction (Windows)

Machine language instruction (MacOS)
Software Development Kit

- JDK (Java Development Kit) editions
  - Java Standard Edition (J2SE): for client-side standalone applications
  - Java Enterprise Edition (J2EE): for server-side applications such as Java servlets and Java Server Pages.
  - Java Micro Edition (J2ME): for applications in mobile devices

- Java versions
  - JDK 1.0 (January 23, 1996)
  - JDK 1.1 (February 19, 1997)
  - J2SE 1.2 (December 8, 1998)
  - J2SE 1.3 (May 8, 2000)
  - J2SE 1.4 (February 6, 2002)
  - J2SE 5.0 (September 30, 2004)
  - Java SE 6 (December 11, 2006)
  - Java SE 7 (July 28, 2011)
  - Java SE 8 (March 18, 2014)
Eclipse

• What is Eclipse?
  ▪ Powerful Integrated Development Environment (IDE)
  ▪ Supporting multiple programming languages
  ▪ Very powerful debugger
  ▪ Allows for easy breakpoint setting
    • Allows for traces of program operation
    • Full view of array values and references
  ▪ Similar to development tools used in industry