Handout 1

Introduction to Algorithms, Data Structures and Java programming language.

Algorithm is a computational method used for solving a problem.

Algorithm design is a part of the process of solving a problem using a computer. Involves:

1. Identifying problem input and output of the problem (analysis)
2. Developing an algorithm for generating the output from the input (algorithm design)
3. Implementing the algorithm in a computer language (coding), i.e. writing a program
4. Verifying the correctness of the program by testing and debugging it.

The term Data structure refers to the logical organization of the information in the computer memory.

The representation of the data is important because it affects the choice of an algorithm for a particular problem, as well as the performance characteristics of an algorithm.

We will use JAVA - a language for Object-Oriented Programming

Object-Oriented Programming - is not the subject of this course.
It is a powerful and popular design and programming technique

Some terminology:

- object – entities of the application domain e.g. person, place or thing (a noun)
- class – a template describing objects of the same type
- method - an action performed by an object (a verb)

- Objects have both data and methods
- Objects of the same class have the same data elements and methods
- Objects send and receive messages to invoke actions

Example: class Automobile.

Data Items: manufacturer’s name, model name, year made, color, number of doors, size of engine

Methods:
- Define data items (specify manufacturer’s name, model, year, etc.)
- Change a data item (color, engine, etc.)
- Display data items
- Calculate cost
Java Program – is a collection of classes.
Example: a very simple Java program called CircleCalculation.java
- Consists of a single class called CircleCalculation with a single method called main.

```java
/*Program to determine area of a circle.
*Author: Jane Q. Programmer.
*Email Address: janeq@somemachine.etc.etc.
*Programming Assignment 2.
********************************************************************/

public class CircleCalculation
{
    public static void main(String[] args)
    {
        double radius; //in inches
        double area; //in square inches

        System.out.println("Enter the radius of a circle in inches:");
        radius = SavitchIn.readLineDouble();
        area = 3.14159 * radius * radius;

        System.out.println("A circle of radius " + radius + " inches");
        System.out.println("has an area of " + area + " square inches.");
    }
}
```

Programming Language Concepts

Programs are written in a high level programming language.
Computers only understand machine instructions that are sequences of 0 and 1’s.

- Machine Language
  - least natural language for humans, most natural language for hardware
  - just 0s and 1s
  - directly understood by hardware
  - not portable (hardware dependent)

- High-Level Language
  - closer to natural language
  - words, numbers, and math symbols
  - not directly understood by hardware
  - “portable” (hardware independent) source code
  - Java, C, C++, COBOL, FORTRAN, BASIC, Lisp, Ada, etc.
Getting from High-Level Language Source to Machine Code

**Compiling a program** - translating from a high-level language source code to machine (object, or executable) code.

**Compiler** - a program that translates source code to machine (object, or executable) code.

**Interpreter** – a program that translates source code instructions into machine code one by one as it executes the program.

**JAVA Program Translation**
- Involves Both Compilation and Interpretation
- Java Compiler (in Sun’s Java Development Kit (JDK) program called **javac**) generates Intermediate Code called “**Byte Code**”
  - Byte Code is the language of the so-called **Java Virtual Machine**.
    - Byte Code is
      - low-level code – easily translatable into machine language
      - **hardware independent**
      - invisible to Java programmer
- An interpreter (program called **java**) translates from byte code to hardware-specific machine code

[Diagram of Java Program Execution]

Java Program (**hello.java**)  

<table>
<thead>
<tr>
<th>Input for Java Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Compiler (<strong>javac</strong>)</td>
</tr>
<tr>
<td>Byte-Code Program (<strong>hello.class</strong>), Intermediate Code</td>
</tr>
<tr>
<td>Byte Code Interpreter (<strong>java</strong>)</td>
</tr>
<tr>
<td><strong>Output of Java Program</strong></td>
</tr>
</tbody>
</table>

**Created by T. Babaian, 08/27/2002 9:07 PM**