Chp 4: Console Input/Output

- Console Input/Output
  - In Java, three special ways to perform console I/O:
    - System.in: standard input
    - System.out: standard output
    - System.err: standard error
  - Console output:
    - Use stream System.out
  - Console input:
    - Use stream System.in
    - Use the class Scanner

- Console Output: print() & println()
  - java.io.PrintStream class
    - print():
      - prints the value of the argument in the specified data type
    - println():
      - same as print() but print out a new line
    - syntax:
      - System.out.print(Output [+ Output]);
      - System.out.println(Output [+ Output]);
    - where Output can be any data type of String, char, int,
      float or double, etc.

- Importing Packages and Classes
  - Basics:
    - Packages are used to group classes
      - e.g.

```
       package test.mypackage;
       public class Class1 {
           public static double method1() {底
       }
```

    - package name informs the compiler the path/directory that
      contains the classes in the package
    - In Java, every class belongs to a package.
    - If we do not specify a package when we define the classes,
      the classes will be part of the default package
  - Example packages in Java
    - java.lang
      - Java Language Package
        contains the core Java classes such as String, Math, etc.
        (automatically imported by compiler)
    - java.io
      - Java Input/Output Package
        for input and output streams and files.
        (other packages need import, e.g., Scanner)
    - java.text
      - Java Text Package
        for manipulating information
    - java.util
      - Java Utilities Package
        contains many utilities such as Scanner
  - The import Statement
    - syntax (at the beginning of the class file):
      import Package_Name.Class_Name;
      import Package_Name.*;
    - e.g.,
      import java.io.IOException;
      import java.io.*;

- Console Input: Using the Scanner Class
  - Basics:
    - available from Java 2 Ver 5
    - usage:
      - for reading data from various sources: keyboard, files, etc.
      - parsing strings into tokens and words (Chp 11)

```
      - from the package java.util
    - To use the Scanner class
      1. import java.util.Scanner;
      2. Create a Scanner object
      3. Use the Scanner methods to read console (keyboard) input
         - for single token:
           - boolean nextBoolean()
           - byte    nextByte()
           - short   nextShort()
           - int     nextInt()
           - long    nextLong()
           - float   nextFloat()
           - double  nextDouble()
           - String  next()
             - Returns the next input token as a String value
         - for one or more tokens:
           - String  nextLine()
             - Returns all input *remaining* on the current line as a String value
    - Can you input more than one value (or token) in a line?
      - e.g.,
          int data1 = sc.nextInt();
          int data2 = sc.nextInt();
          int data3 = sc.nextInt();
        - read user input data: 10__20__30
          -> when enter with keyboard, must separate by blank spaces / white spaces
    - Note:
      if enter incorrect input data, runtime error (or exception)
      will occur (see Chp 13)
    - About nextLine()
      - It reads the remainder of a line of text wherever the last
        keyboard reading left off
        -> \n is left off by single-token reading method!!!!
      - Useful solution:
        -> Place a nextLine() method after reading a single token
           IF user will press an enter after that input

  - [Not required in this course, but useful to know]
    - Output/Input Dialog Box
      - based on Graphical User Interface (GUI)
      - import javax.swing.JOptionPane
      - use class JOptionPane
    - Formatted Console Output
      - Java supports formatted output with
        - DecimalFormat Class
        - NumberFormat Class
        - System.out.printf() Method
          (like C programming language)
```