Chp 3: Data

- Memory and Variables
  - Syntax: Declare a variable
    1. DataType
    2. Variable Name
    3. Value (for Initialization, optional)

    e.g., int variable1 = 10 ;

- Data Types
  - A data object is of a certain type that requires the specified memory storage size
  - Category:
    - Primitive Data Types
      - Boolean: true or false
        - 1 byte
      - Character
        - 2 bytes
      - Numerical
        - floating-point:
          - for numerical value with fractional values
          - float (4 bytes) and double (8 bytes)
          - may not be precise; significant digits
        - integer type:
          - for integers
          - byte, short, int, and long (1,2,4,8 bytes)
    - Reference Data Types
      - store an address of an object so that we can use the address to refer to the object data
      - e.g., String data type
  - A data object in a Java program can be
    - a constant
    - a variable
  - Java reserves the necessary data storage locations depending on the data type of the data object

- Literals
  - values (associated with data type) used directly in program
  - five types of literals
    - Integer literals
      - e.g. 100,-256
    - Floating-point literals
      - e.g. 2.4, -3.0
    - Character literals
      - e.g. 'a', '+'
    - Boolean literals
      - e.g. true, false
    - String literals
      - e.g. "Hello Students"
  - Integer Literals
    - default: int type
    - append L/l to indicate long, e.g., 10L
    - octal:
      - leading 0
      - e.g.,010 is 8
    - hexidecimal
      - leading 0x or 0X
      - e.g., 0x10 is 16
  - Floating-point Literals
    - default: double type
    - append F/f to indicate float
    - append D/d to indicate double (but redundant)
    - can use scientific notations, e.g., 1.23456e-2
  - Character Literals
    - First 128 characters for ASCII
e.g., 0x41 = 65 = 'A'
    - In Java: 2 bytes

```
      - can store UNICODE including Chinese, Korean, etc.
    - Escape Sequence
      - non-printable control characters, e.g.,
        \n for newline
        \a for alarm bell
      - special character for String, e.g,
        \\ for backslash itself
        \" for double quote
        \' for single quote

  - Identifiers
    - used to name things such as variables, constants,
      classes and packages.
    - Rules for naming identifiers
      - must start with a letter, an underscore character (_)
        or a dollar sign ($)
      - may contain only letters, digits (0,..,9), and the
        underscore character (_)
      - case sensitive
      - cannot contain a space, or any other characters such as
        a dot (.) or an asterisk (*)
      - cannot be a reserved word or keyword
      - does not have any length limit

  - Constants
    - object whose value is unchanged throughout program execution
    - syntax:
      final Type CONSTANT_NAME = Value;
    - e.g.,
      final double PI = 3.14159;
    - If you place the constant outside the main() method
      static final PI = 3.14159 ;
      need to add keyword "static"
    - Why use constants?
      - improves the readability of the program
      - avoid potential programming mistakes; good programming practice

  - Variables
    - basics:
      - data objects that may change and be assigned values as
        the program runs
      - each variable has a name
      - variable names are case sensitive
      - use meaningful names (good programming practice)
      - cannot be any of the keywords in Java
    - declaration
      - create a variable
      - Variables are declared by declaration statements
      - declaration can be done with or without initialization
      - can declare multiple variables in one statement, e.g.,
        int a , b ;
      - To improve readability of the program, declare initialized
        and un-initialized variables in separate declaration statements
      - During compilation, a memory location of suitable size is
        assigned for each variable
      - A variable must be declared before it is used
```