SE 102 Abstract Data Type and Problem Solving

Lecture 1

Asst Prof. Pree Thiengburanathum, PhD.

pree.t@cmu.ac.th

Office room 415-1

CAMT
College of Arts, Media and Technology
Chiang Mai University

Agenda

- ~~Course Syllabus~~
- Java Fundamentals
  - Compiling Java (new JDK and JRE)
  - Comment, identifiers and variable names
  - Primitive data types
  - Method
  - One dimensional array and multi-dimensional array

Course Syllabus

- Can be retrieved from Moodle

Compiling a Java program

- Eclipse or VSCode as integrated development environment (IDE)
  - Download and install Eclipse IDR for Java Developers
    - www.eclipse.org, https://code.visualstudio.com/
- Compile using *javac* and run via command line
  - https://introcs.cs.princeton.edu/java/15inout/windows-cmd.html
- Compile java source code online (covid 19 effected student)
  - https://www.compilejava.net/
  - https://www.jdoodle.com/online-java-compiler
  - http://rextester.com/l/java_online_compiler

Java comments

Each comment line in Java begins with
"/**"
and ends with
"*/

A one-line comment begins with "//“

```
/**
 * Hello1 --- program to print "Hello World".
 * @author    Pree T.
 */
public class Hello1 {
 :
}
```

Java comments

```java
// Counts the number of occurrences of the provided name
// in the given list of names.
//
// Pre:
// - The list of names must not be null, otherwise an IllegalArgumentException
//   will be thrown.
// - The target name must not be null, otherwise an IllegalArgumentException
//   will be thrown.
//
// Post:
// - Returns the number of times targetName appears in names as a a positive int.
public static int numberOfOccurrences(List<String> names, String targetName) {
    int count = 0;
    for (String name : names) {
        if (name.equals(targetName)) }
            count += 1;
        }
    }
    return count;
}
```

Identifiers and variable names in Java

- *Identifiers* are the names of variables, methods, classes, packages and interfaces. Unlike literals they are not the things themselves, just ways of referring to them.
- In the HelloWorld program, HelloWorld, String, args, main and println are identifiers.
- My Variable // Contains a space
- 9pins // Begins with a digit
- a+c // The plus sign is not an alphanumeric character
- testing1-2-3 // The hyphen is not an alphanumeric character
- O'Reilly // Apostrophe is not an alphanumeric character
- OReilly_&_Associates // ampersand is not an alphanumeric character

Data Type

- Data Type
  - a data type is a set of values together with an associated collection of operators for manipulating those values.
- 2 types of data type
  - ~~Primitive data type i.e. int, double, char, boolean, floating point~~
  - Defined data type i.e. array, list

Java operators

Operator	Description	Example
+ (Addition)	Adds values on either side of the operator.	A + B will give 30
- (Subtraction)	Subtracts right-hand operand from left-hand operand.	A - B will give -10
* (Multiplication)	Multiplies values on either side of the operator.	A * B will give 200
/ (Division)	Divides left-hand operand by right-hand operand.	B / A will give 2
% (Modulus)	Divides left-hand operand by right-hand operand and returns remainder.	B % A will give 0
++ (Increment)	Increases the value of operand by 1.	B++ gives 21
-- (Decrement)	Decreases the value of operand by 1.	B-- gives 19

Primitive data type **int**

Integers can be both positive and negative

A subset of real number

- The / operator denotes integer division
  - a/b evaluates to a divided by b, discarding any remainder
  - 5/2 evaluates to 2
  - −23/6 evaluates to −3
  - 4/43 has the value 0
- The expression a%b evaluates to the remainder of a divided by b
  - 5%2 has the value 1
  - −23%3 has the value −2

Primitive data type **double**

- The values are decimal numbers in the range -1.7×10^{308}... 1.7×10^{308} with 14 significant digits of accuracy.

- The division operator (/) denotes decimal or floating point division rather than integer division; so 5.0/2.0 has the value 2.5 but 5/2 has the value 2

Primitive data type **char**

- Type *char* is the set of all characters found on the standard keyboard
- Java uses the Unicode character set and allocates two bytes of memory for each character.
- Using two bytes allows 65,536 characters.
- In java, A value of type char is enclosed in single quotes:
- '5'is a value of type char,
- "5"is a string literal, and
- 5 is an integer.

Java operator precedence

- int x = 4 + 3 * 5;

- 1. (4 + 3) * 5 == 35
- 2. 4 + (3 * 5) == 19

- In the absence of parentheses, which choice is appropriate?
- In the case of Java, multiplication takes precedence over addition; therefore, x will get the value 19.
- http://www.cs.bilkent.edu.tr/~guvenir/courses/CS101/op_precedence.html

Primitive data type **boolean**

- Type boolean has two values: true and false
- The associated operators:
- '&&' --'and'
- '||' --'or'
- '!' --'not'

The truth table

x	y	x && y (and)	x || y (or)	!x (not)
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Methods

- A *method* is a named sequence of instructions that are grouped together to perform a task.

- Methods enable the programmer to organize various tasks into neat manageable independent *bundles* of code

Methods

- Every Java application must have a *main* method
- The execution of every Java application begins with the main method.
- Other methods that we have used are print(...), println(...),

Java's Pre-defined Methods

- Imagine a mathematical "*black box*" that works in such a way that whenever you supply a number to the box

- The box gives or returns the positive square root of that number.

Java's Pre-defined Methods

- A similar mechanism that accepts *two* numbers, the *length* and *width* of a rectangle, and returns the area of the rectangle.

4 formats of Method

- *void* method1 ()
- *void* method1(arg1,arg2.., argn)
- *returned type* method1()
- *returned type* method1(arg1,arg2.., argn)

Method example

```
/** the snippet returns the minimum between two numbers */

public static int minFunction(int n1, int n2) {
   int min;
   if (n1 > n2)
      min = n2;
   else
      min = n1;

   return min;
}
```

Method example (cont.)

```
public class ExampleMinNumber {


  public static void main(String[] args) {
    int a = args[0];
   int b = 6;
    int c = minFunction(a, b);\
     System.out.println("Minimum Value = " + c);
  }
```

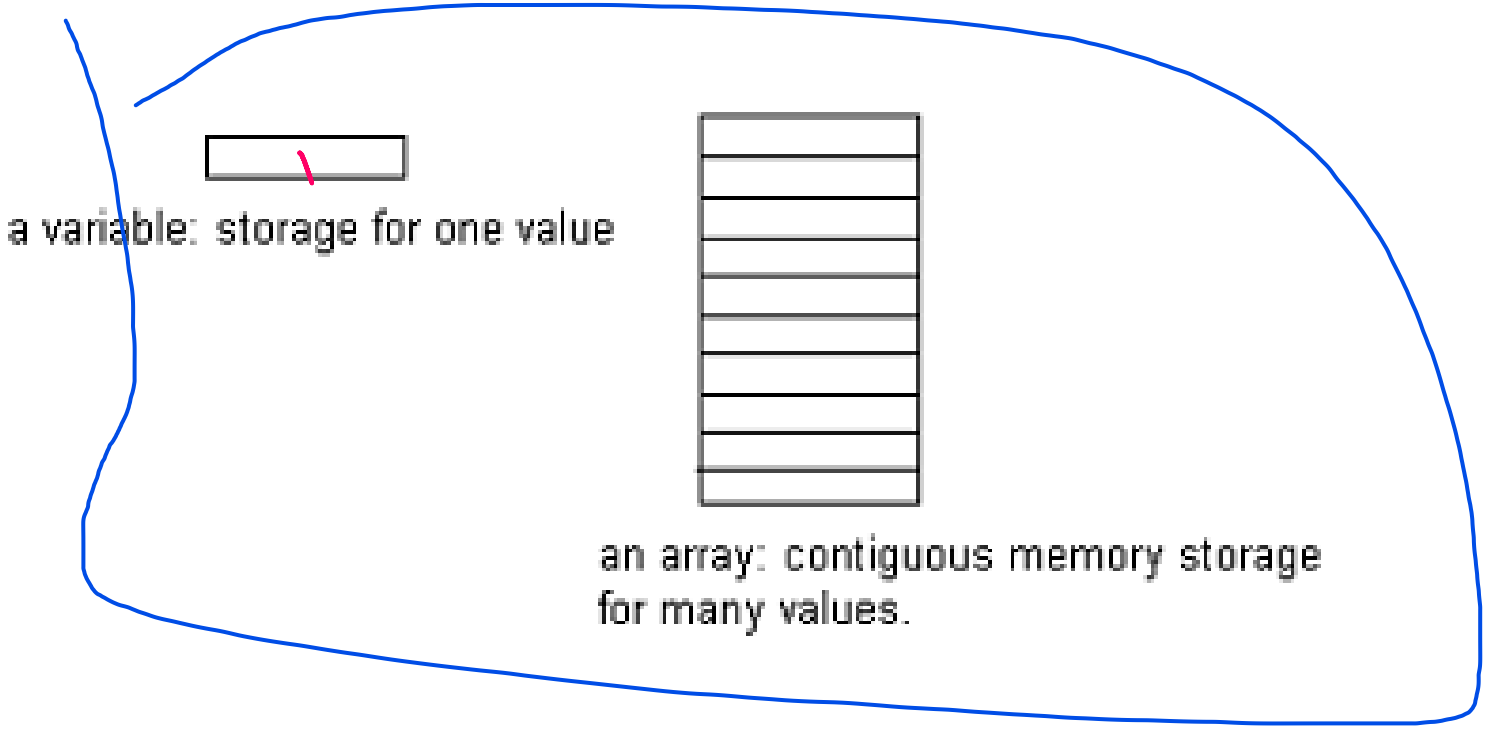
DEMO

Array

- An array is a collection of **elements**, **items**, or **values** that have the same data type.

- Unlike the variables of previous programs, an array can store *more than one* value.

A variable in contrast to an array



Array Instantiation

- Once an array is created, its length is *fixed*.
- The length of an array cannot be *altered*.
- If variable x refers to an array, then x.length gives the number of memory cells allocated to the array.

Creating Arrays

- General syntax for declaring an array:

```
Base_Type[] Array_Name = new Base_Type[Length];
```

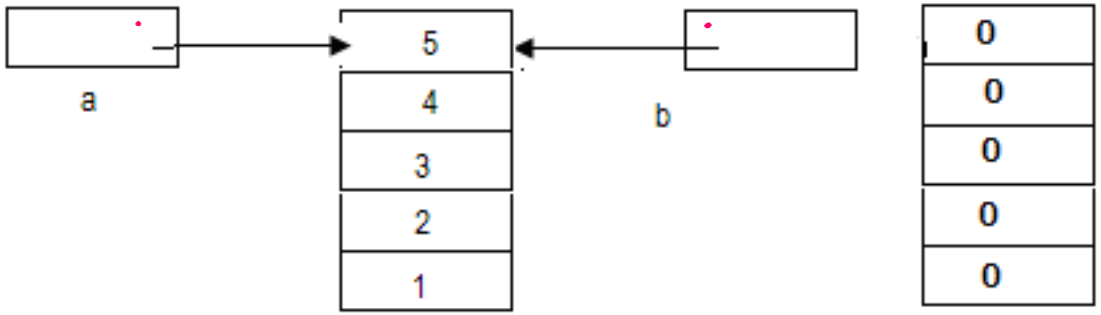
- Examples:
  80-element array with base type `char`:
  ```
  char[] symbol = new char[80];
  ```

  100-element array of `doubles`:
  ```
  double[] reading = new double[100];
  ```

  70-element array of `Species`:
  ```
  Species[] specimen = new Species[70];
  ```

The = operator

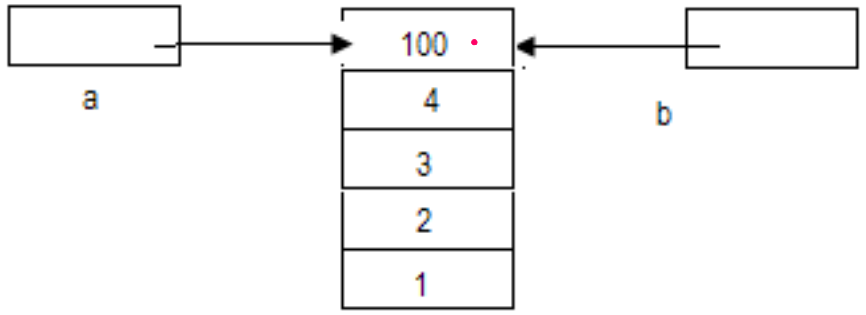
- b=a



- After the assignment b=a

The = operator

- B[0] =100;



- After executing the assignment b=a, the references a and b both refer to the same memory and any changes to a[i] affect b[i];

One dimension array

// **Declaration of allocating memory to an array**
int iarr[] = new int[3];

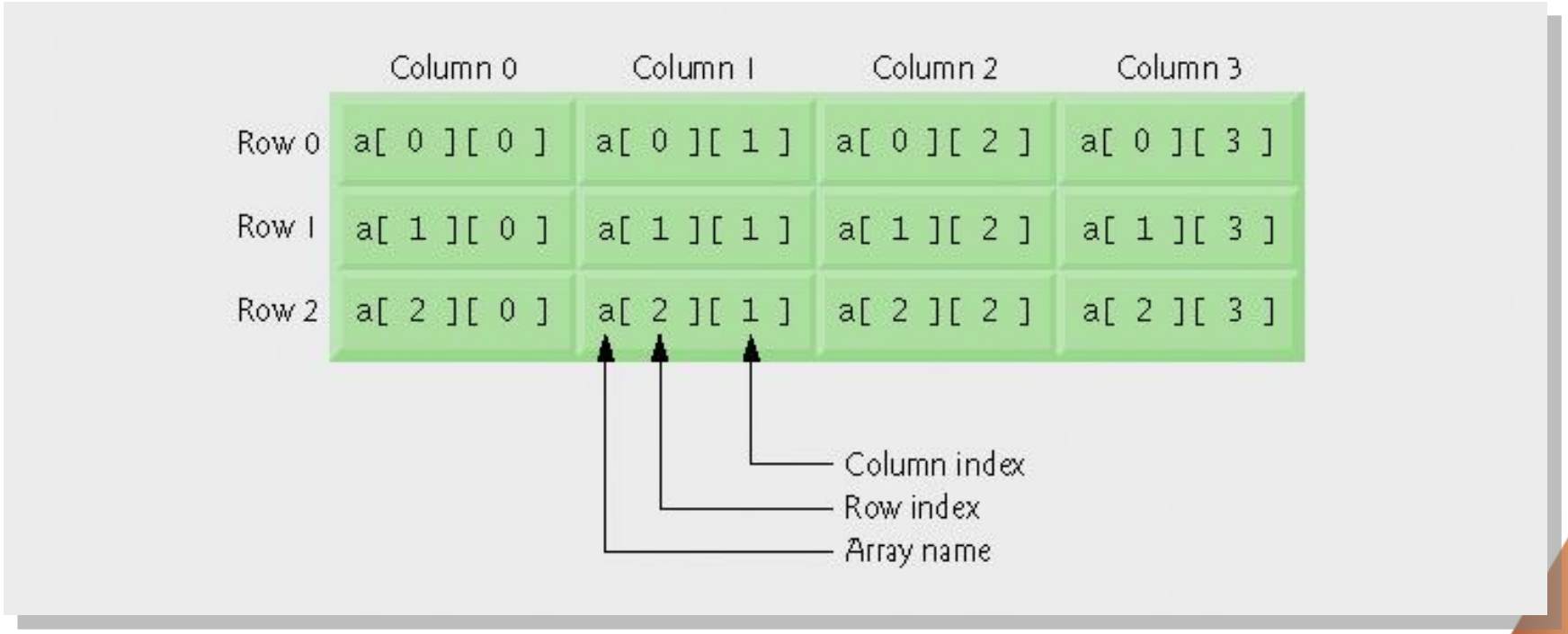
// **Initializing elements**
iarr[0] = 1;
iarr[1] = 2;
iarr[2] = 3;

One dimensional array

```
//Display array elements
System.out.println(iarr[0]);
System.out.println(iarr[1]);
System.out.println(iarr[2]);


// Or Use for loop to display elements
 for (int i = 0; i < iarr.length; i = i + 1)
{
      System.out.print(iarr[i]);
      System.out.print(" ");
 }
```

Multidimensional Arrays



Two-dimensional array with three rows and four columns.

Creating Two-dimensional Arrays

- Can be created dynamically
- 3-by-4 array

```
int b[][];
b = new int[ 3 ][ 4 ];
```

- Rows can have different number of columns

```
int b[][];
b = new int[ 2 ][ ];
// create 2 rows
b[ 0 ] = new int[ 5 ];
// create 5 columns for row 0
b[ 1 ] = new int[ 3 ];
// create 3 columns for row
```

Q&A

CAMT
College of Arts, Media and Technology
Chiang Mai University