Beyond 1-Dimensional Arrays

- 2D or high-dimensional arrays are widely used.
  - For example, a colorful image is represented by three 2D arrays (R, G, B).
- We can create a 2D \( T \)-type array with 4 rows and 3 columns as follows:

```
1 ...
2     int rowSize = 4; // row size
3     int colSize = 3; // column size
4     T[][] M = new T[rowSize][colSize];
5 ...
```
Case (c) shows that we can create a 2D array by enumeration.
Reality

```java
int[][] triangleArray = {
    {1, 2, 3, 4, 5},
    {2, 3, 4, 5},
    {3, 4, 5},
    {4, 5},
    {5}
};
```
Example

```
... int[][] A = {{1, 2, 3}, {4, 5}, {6}};

// conventional for loop
for (int i = 0; i < A.length; i++) {
    for (int j = 0; j < A[i].length; j++)
        System.out.printf("%2d", A[i][j]);
    System.out.println();
}

// for-each loop
for (int[] row: A) {
    for (int item: row)
        System.out.printf("%2d", item);
    System.out.println();
}
...
```

1Thanks to a lively discussion on January 31, 2016.
Exercise: Matrix Multiplication

Write a program which determines \( C = A \times B \) for the input matrices \( A_{m \times n} \) and \( B_{n \times q} \) for \( m, n, q \in \mathbb{N} \).

- You may use the formula

\[
c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}
\]

where \( a_{ik}, i = 1, 2, \ldots, m \) is a shorthand for \( A \) and \( b_{kj}, j = 1, 2, \ldots, q \) for \( B \).

- Time complexity: \( O(n^3) \) (Why?)
class Lecture6 {

   "Methods"

}

// keywords:
return
Methods

- Methods can be used to define reusable code, and organize and simplify code.
- The idea of function originates from math, that is, 
  \[ y = f(x), \]
  where \( x \) is the input parameter\(^2\) and \( y \) is the function value.
- In computer science, each input parameter should be declared with a specific type, and a function should be assigned with a return type.

\(^2\)Recall the multivariate functions. The input can be a vector, say the position vector \((x, y, z)\).
\(^3\)Aka procedures and functions.
Example: max

Define a method

```
public static int max(int num1, int num2) {
    int result;
    if (num1 > num2)
        result = num1;
    else
        result = num2;
    return result;
}
```

Invoke a method

```
int z = max(x, y);
```
... modifiers returnType methodName(listOfParameters) {
    // method body
}
...

- **modifiers** could be **static** and **public** (for now).
- **returnType** could be primitive types and reference types.
  - If the method does not return any value, then the return type is declared **void**.
- **listOfParameters** is used to indicate the method inputs, each separated by commas.
  - Note that a method could have no input.\(^4\)
- The method name and the parameter list together are called the **method signature**.\(^5\)

\(^4\)For example, `Math.random()`.

\(^5\)**Method overloading** depends signatures. We will see it soon.
More Observations

• There are alternatives to the method `max()`:

```java
... public static int max(int num1, int num2) {
    if (num1 > num2) {
        return num1;
    } else {
        return num2;
    }
}
...
```

```java
... public static int max(int num1, int num2) {
    return num1 > num2 ? num1 : num2;
}
...
```
“All roads lead to Rome.”
– Anonymous

“但如你根本並無招式，敵人如何來破你的招式？”
– 風清揚，笑傲江湖。第十回。傳劍
The return Statement

- The return statement is the end point of the method.
- A callee is a method invoked by a caller.
- The callee returns to the caller if the callee
  - completes all the statements (w/o a return statement, say main());
  - reaches a return statement;
  - throws an exception (introduced later).
- As you can see, the return statement is not necessarily at the bottom of the method.\(^6\)
- Once one defines the return type (except void), the method should guarantee to return a value or an object of that type.

\(^6\)Thanks to a lively discussion on November 22, 2015.
Bad Examples

...  
  public static int foo1() {  
      while (true);  
      return 0; // unreachable code  
  }  

  public static int foo2(int x) {  
      if (x > 0) {  
          return x;  
      }  
      // what if x < 0?  
  }  
...
Method Invocation

- Note that the input parameters are sort of variables declared within the method as placeholders.
- When calling the method, it’s the obligation of callers to provide arguments in order, number, and compatible type, as defined in the method signature.
• In Java, method invocation uses **pass-by-value**.

• When the callee is invoked, the **program control (pc)** is transferred from the caller to the callee.

• For each method invocation, JVM pushes a **frame** which stores necessary information in the **call stack**.

• The caller resumes its work once the callee finishes its routine.
(a) The main method is invoked.

(b) The max method is invoked.

(c) The max method is being executed.

(d) The max method is finished and the return value is sent to k.

(e) The main method is finished.
Variable Scope

- A variable scope refers to the **region** where a variable can be referenced.
- A pair of balanced curly braces defines the variable scope.
- In general, variables can be declared in **class level**, **method level**, or **loop level**.
- We **cannot** duplicate the variables whose names are identical in the same level.
public class ScopeDemo {

    public static int x = 10; // class level

    public static void main(String[] args) {

        System.out.println(x); // output 10

        int x = 100; // method level, aka local variable
        x++;
        System.out.println(x); // output 101
        addOne();
        System.out.println(x); // output ?
    }

    public static void addOne() {
        x = x + 1;
        System.out.println(x); // output ?
    }
}

A Math Toolbox: **Math** Class

- The **Math** class provides basic mathematical functions and 2 global constants **Math.PI**\(^7\) and **Math.E**\(^8\).
- All methods are **public** and **static**.
  - For example, `max`, `min`, `round`, `ceil`, `floor`, `abs`, `pow`, `exp`, `sqrt`, `cbrt`, `log`, `log10`, `sin`, `cos`, `asin`, `acos`, and `random`.
- Full document for **Math** class can be found [here](#).
- You are expected to read the document!

---

\(^7\) The constant \(\pi\) is a mathematical constant, the ratio of a circle’s circumference to its diameter, commonly approximated as 3.141593.

\(^8\) The constant \(e\) is the base of the natural logarithm. It is approximately equal to 2.71828.
• Methods with the same name can coexist and be identified by the method signatures.

```java
...  
public static int max(int x, int y) { ... }  
// different numbers of inputs
public static int max(int x, int y, int z) { ... }  
// different types
public static double max(double x, double y) { ... }
...
```