

# Software Practice 1

## Basic Grammar

- **Basic Syntax**
- **Data Type**
- **Loop Control**
- **Making Decision**

Prof. Joonwon Lee

T.A. Jaehyun Song  
Jongseok Kim (42)

T.A. Sujin Oh  
Junseong Lee (43)

# Java Program

```
//package details
```

```
public class ClassName {  
    Type classVariableName;  
    public void methodName(arguments) {  
        Type localVariableName;  
    }  
    public static void main(String args[]) {  
        statements;  
    }  
}
```

# Basic Syntax

- **Case Sensitivity** – Java is case sensitive, which means identifier **Hello** and **hello** would have different meaning in Java.
- **Class Names** – For all class names the first letter should be in Upper Case. If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.

**Example:** *class MyFirstJavaClass*

- **Method Names** – All method names should start with a Lower Case letter. If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.

**Example:** *public void myMethodName()*

# Basic Syntax (cont'd)

- **Program File Name** – Name of the program file should exactly match the class name.

**Example:** Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as *'MyFirstJavaProgram.java'*

- **public static void main(String args[])** – Java program processing starts from the main() method which is a mandatory part of every Java program.

# Java Identifiers

- All Java components require names. Names used for classes, variables, and methods are called **identifiers**.
- In Java, there are several points to remember about identifiers. They are as follows –
  - All identifiers should begin with a letter (A to Z or a to z), currency character (\$) or an underscore (\_).
  - After the first character, identifiers can have any combination of characters.
  - A keyword cannot be used as an identifier.
  - Most importantly, identifiers are case sensitive.
  - Examples of legal identifiers: age, \$salary, \_value, \_\_1\_value.
  - Examples of illegal identifiers: 123abc, -salary.

# Java Modifiers

- Like other languages, it is possible to modify classes, methods, etc., by using modifiers. There are two categories of modifiers –
  - Access Modifiers – default, public , protected, private
  - Non-access Modifiers – final, abstract, strictfp
- We will be looking into more details about modifiers in the 4<sup>th</sup> week.

# Java Keywords

- Keywords are reserved words in Java. These reserved words may not be used as constant or variable or any other identifier names.
- You can check every keywords on following link
  - [https://www.tutorialspoint.com/java/java\\_basic\\_syntax.htm](https://www.tutorialspoint.com/java/java_basic_syntax.htm)

abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
...			

# Primitive Data Types

- **Kinds of values that can be stored and manipulated**

**byte : 8-bit signed two's complement integer**

**boolean : truth value (true or false)**

**short : 16-bit signed two's complement integer**

**int : 32-bit signed two's complement integer**

**long : 64-bit signed two's complement integer**

**float : single-precision 32-bit IEEE 754 floating point**

**double : double-precision 64-bit IEEE 754 floating point**

**char : single 16-bit Unicode character**



# Variables

- A variable provides us with named storage that our programs can manipulate. Each variable in Java has a specific type, which determines the size and layout of the variable's memory;

**Form: TYPE NAME;**

**Example: String foo;**

# Array

- Java provides a data structure, the **array**, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- Syntax

```
Type[] arrayRefVar; // preferred way.
```

```
Type arrayRefVar[]; // works but not preferred way.
```

# Array Declaration

- Java does not provide the dynamic size array which is able to resize the array dynamically. So, size of the array has to be specified before accessing the array.
- Syntax

```
Type[] arrayRefVar = new Type[12];
```

# Output to Console

- **System.out** //standard out
- **System.out.println(data);**
- **System.out.print(data);**

**System.out.println("Hello Java!");**

**System.out.print("Hello Java!");**

# Input from Keyboard

- **System.in** //standard in

```
int b;
```

```
b = System.in.read(); // read by ascii
```

```
System.out.println(b);
```

# Input from Keyboard

- (소곤소곤) Off the record

```
import java.util.Scanner;  
Scanner s = new Scanner (System.in);  
while (s.nextInt ()) {  
    ...  
}
```

# Basic Operator

- Java provides a rich set of operators to manipulate variables. We can divide all the Java operators into the following groups –
  - Arithmetic Operators
  - Relational Operators
  - Bitwise Operators
  - Logical Operators
  - Assignment Operators

# Loop Control

- **while loop**
- **for loop**
- **do ... while loop**
- **Loop control statements**
  - break;
  - continue;
- **Enhanced for loop in Java**
  - This is mainly used to traverse collection of elements including arrays.



# Enhanced for Loop

- As of Java 5, the enhanced for loop was introduced. This is mainly used to traverse collection of elements including arrays.
- ```
for(declaration : expression) {  
    // Statements  
}
```

# Enhanced for Loop Example

```
public class Test {
    public static void main(String args[]) {
        int [] numbers = {10, 20, 30, 40, 50};
        for(int x : numbers ) {
            System.out.print( x );
            System.out.print(",");
        }
        System.out.print("\n");
        String [] names = {"James", "Larry", "Tom", "Lacy"};
        for( String name : names ) {
            System.out.print( name );
            System.out.print(",");
        }
    }
}
```

# Decision Making

- **if statement**
- **if ... statement**
- **Nested if statement**
- **switch statement**
- **The Conditional Operator ( ? : )**
  - Expr1 ? Expr2 : Expr3;

# [Lab – Practice #1]

## ▪ Number System Converter

- Input decimal number from keyboard
- Convert input to binary, octal, hexadecimal number
- Print to console the results.

Ex)

Input decimal number : 2017

Binary number : 11111100001

Octal number : 3741

Hexadecimal number : 7e1

# [Submit]

## ■ Upload to i-Campus

- Compress your .java file to zip file
- File name: studentID\_lab01.zip

## ■ Due date

- Today 23:59:59
  - Class 42 (3/12 Monday)
  - Class 43 (3/14 Wednesday)
- Penalty: **-10%** of each lab score per **one day**