Java Basics

Lab 2

TA : Changmin Jeon, Minji Kim, Hyunwoo Jung,
Hyunseok Oh, Jingyu Lee, Seungwoo Cho

SEOUL NATIONAL UNIVERSITY

Announcement

- You should finish the lab practice and submit your job to eTL before the next lab class starts(Wednesday, 6:30 PM).
- The answer of the practice will be uploaded after the due.

Overview

- Java basic review
    - Arrays
    - if-else / ternary / switch
    - while / for / foreach
- Practice 1 - Reverse Print
- Practice 2 - Student ID Checker

Java Basic Review: Arrays

Main Function

```
String[] emptyArr = new String[5];
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);
cars[0] = "Opel";
System.out.println(cars[0]);
System.out.println(cars.length);
```

Output

```
Volvo
Opel
4
```

Java Basic Review: if/else Statement

Main Function

```
int time = 22;
if (time < 10) {
  System.out.println("Good morning.");
} else if (time < 20) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
}
```

Output

```
Good evening
```

Java Basic Review: Ternary Operator

Main Function

```
int a = 10, b = 20;
String result = a > b ? "a is greater" : "b is greater";
System.out.println(result);
```

Output

```
b is greater
```

Java Basic Review: `switch` Statement

Main Function

```
int score = 2;
switch (score) {
  case 2:
    System.out.println("Your score is 2.");
    break;
  case 3:
    System.out.println("Your score is 3.");
    break;
  default:
    System.out.println("End of statement.");
}
```

Output

```
Your score is 2.
```

Java Basic Review: while/do-while

Main Function

```
int i = 0;
while (i < 5) { System.out.print(i++ + ","); }
System.out.println();

i = 0;
do { System.out.print(i++ + ","); }
while (i < 5);
System.out.println();
```

Output

```
0,1,2,3,4,
0,1,2,3,4,
```

Java Basic Review: for/for-each

Main Function

```
String[] cars = { "Volvo", "BMW", "Ford", "Mazda"};

for (int i = 0; i < 4; i++) {
  System.out.print(cars[i] + " ");
}
System.out.println();

for (String car : cars) { System.out.print(car + " "); }
System.out.println();
```

Output

```
Volvo BMW Ford Mazda
Volvo BMW Ford Mazda
```

Practice 1 : Array Printer

- Write a program which inputs strings and outputs in the opposite order.
    - Get the number of input strings
    - Declare a string array
    - Get input strings and put them into the array
    - Print the strings of the array
    - Print the strings of the array in the opposite order

Array Printer 1 - Get String Input

ArrayPrinter.java

```
import java.util.Scanner; // Import scanner

public class ArrayPrinter {

  public static void main(String[] args) {
    // Create a scanner which get inputs from console
    Scanner scanner = new Scanner(System.in);

    // Get the input as "int" type
    int numInput = scanner.nextInt();
    System.out.println(numInput);
  }
}
```

Console

```
3 # Your input
3 # Output
```

Array Printer 2 - Save Strings in an Array

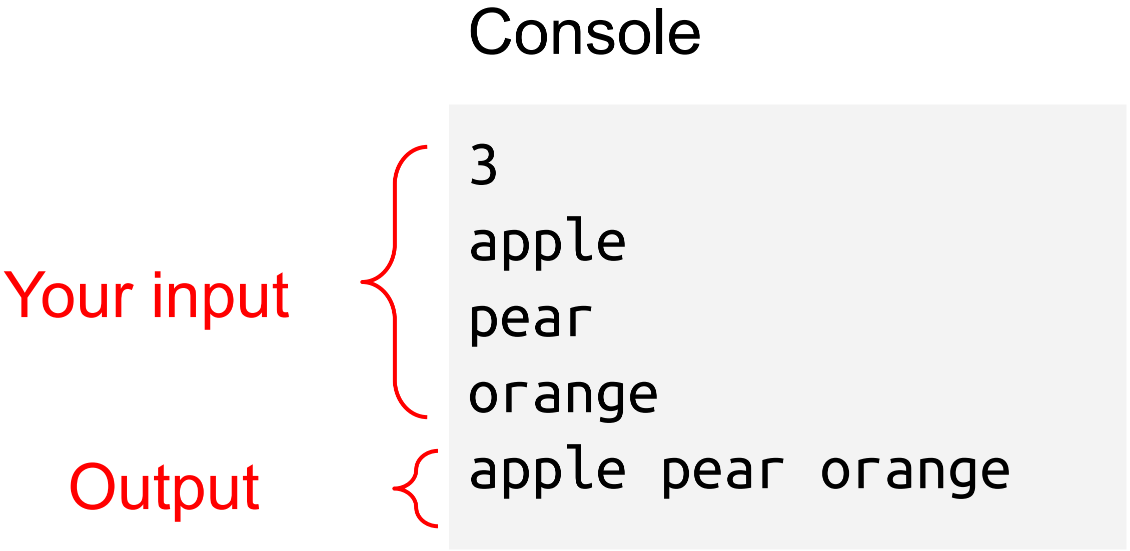
```
...
  String[] arr = new String[numInput];
  ~~System.out.println(numInput);~~

  for (int i = 0; i < numInput; i++) {
   // Get string input at each iteration
   String input = scanner.next();
   arr[i] = input; // Put the input into the array
  }

  // for-each loop: Iterate on each element in the array
  for (String string : arr) {
   System.out.print(string + " ");
  }
  System.out.println(); // Break line
 }
}
```

Add this part

Array Printer 2 - Save String in an Array



Console

```
3
apple
pear
orange
apple pear orange
```

Your input

Output

Array Printer 3 - Reverse Print

```
...
   System.out.print(string + " ");
  }
  System.out.println(); // Break line

  // i = numInput - 1, numInput - 2, ..., 0
  for (int i = numInput - 1; i >= 0; i--) {
   System.out.print(arr[i] + " ");
  }
  System.out.println(); // Break line
 }
}
```

Add this part

Array Printer 3 - Reverse Print

Console

Your input

```
3
apple
pear
orange
```

Output

```
apple pear orange
orange pear apple
```

Practice 2-1 : Student ID Validator

- Write a program which checks whether an input string is a valid student ID (XXXX-XXXXX).
- Input a string from the console and save the string into a variable.
- Check whether the input string is a valid student ID or not, and print a corresponding message.
  a. The length of the input should be 10.
  b. The 5th character of the input should be ‘-’.
  c. All characters of the input but 5th should be digits.

Get nth character in a String

- Use `.charAt` to get a nth character of a string
- Pass an `int` variable as a index of the character you want to get.
- Return type of `.charAt` is `char`.
- `IndexOutOfBoundsException` is thrown if the index argument is negative or not less than the length of this string.

Main Function

```
System.out.println("abcde".charAt(3));
```

Out

```
d
```

Check whether a character is a digit

- Each Java character matches to a number called ASCII code (https://en.wikipedia.org/wiki/ASCII)
- You can check whether a character is a digit or alphabet with ASCII code comparison.
- This boolean expression is `true` if `char` type variable `ch` is a
  - digit: `ch >= '0' && ch <= '9'`
  - non-digit: `ch < '0' || ch > '9'`
  - lower alphabet: `ch >= 'a' && ch <= 'z'`
  - upper alphabet: `ch >= 'A' && ch <= 'Z'`

Student ID Validator 1

Console

```
2018-1234 # Your input
The input length should be 10. # Output
```

Console

```
2018_12345 # Your input
Fifth character should be ‘-’. # Output
```

Console

```
e018-12345 # Your input
Contains an invalid digit. # Output
```

Console

```
2018-12345 # Your input
2018-12345 is a valid student ID # Output
```

Practice 2-2 : Student ID Validator - Refactoring

- Refactor (Make the code clean) student ID checker by
  - moving each validation checking logic into new functions, `isProperLength`, `hasProperDivision`, and `hasProperDigits`.
  - moving top-level `if`/`else` statements into a new function `validateStudentID`.

Practice 2-3 : Student ID Validator - Repeated Input

- Upgrade your student ID checker to get input repeatedly until the input is “exit”.

Student ID Validator 3 - Repeated Input

Console

```
Input   2018-1234
Output  The input length should be 10.
Input   2018_12345
Output  Fifth character should be ‘-’.
Input   ee18-12345
Output  Contains an invalid digit.
Input   2018-12345
Output  2018-12345 is a valid student ID.
```

Submission

- Compress your final StudentIDValidator.java file into a zip file.
- Rename your zip file as 20XX-XXXXX_{name}.zip - for example, 2020-12345_KimMinji.zip
- Upload it to eTL - Lab 2 assignment.
- Your program should contain main function that can be properly executed and print desired outputs.