Java Basics

Lab 01

Announcement

- We will present assignments after each lab class, but only for this time, the assignments will be uploaded later this week.
- The assignment is only given out to help you study. They will not be evaluated.
- Midterm/final exams will contain questions based on the assignments.
- We STRONGLY RECOMMEND you to do the assignments.

Overview

- Java basic review
  - Arrays
  - if-else / ternary / switch
  - while / for / foreach
  - functions
- 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 < 5; 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
```

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

```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

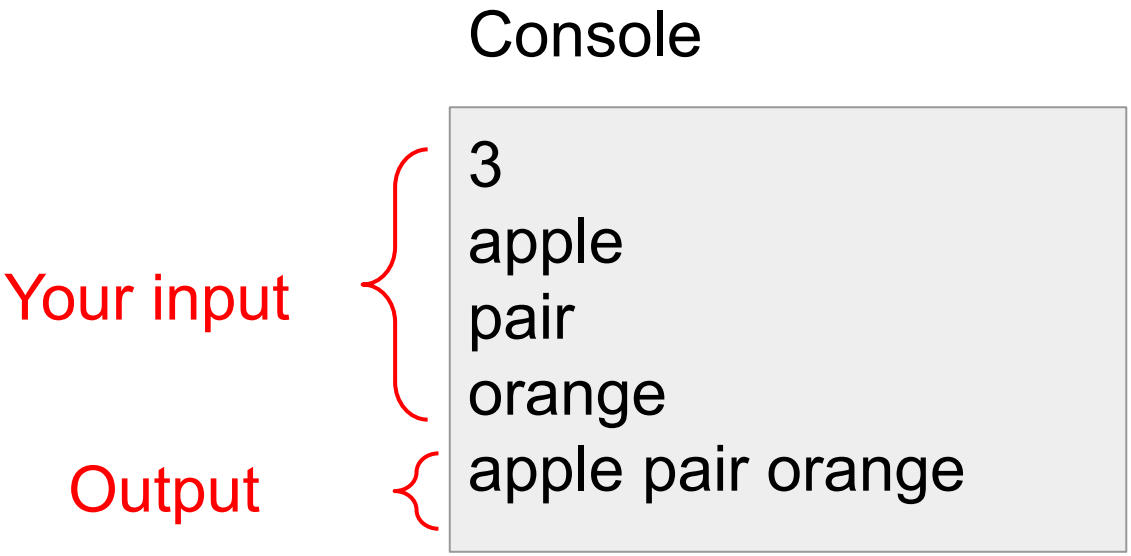
```
...
    String[] arr = new String[numInput];
    System.out.println(numInput);

    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



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
pair
orange
```

Output

```
apple pair orange
orange pair apple
```

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

- Write a program which check an input string is a valid student ID (20XX-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.

Student ID Validator 1

StudentIDValidator.java

```
import java.util.Scanner;

// Validate input student ID (20XX-XXXXX)
public class StudentIDValidator {

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String input = scanner.next();

    // Validate whether the input string length is 10
    if (input.length() != 10) {
      System.out.println("The input length should be 10.");
      return; // “return” is needed to end the function not
to execute the rest of the codes
...
```

Student ID Validator 1

```
...
    Scanner scanner = new Scanner(System.in);
    String input = scanner.next();

    // Validate whether the input string length is 10
    if (input.length() != 10) {
      System.out.println("The input length should be 10.");
      return;

 // Validate whether the 5th character of the input is '-'
    } else if (input.charAt(4) != '-') {
      System.out.println("Fifth character should be '-'.");
      return;
...
```

Student ID Validator 1

```
...
    } else if (input.charAt(4) != '-') {
      System.out.println("Fifth character should be '-'.");

      // Validate whether the characters of the input except 5th
character is digits (0, 1, ..., 9)
    } else {
      for (int i = 0; i < input.length(); i++) {
        if (i != 4) {
          char ch = input.charAt(i);
          if (ch < '0' || ch > '9') {
            System.out.println("Contains an invalid digit.");
          }
        }
      }
    }
    // Validation success
    System.out.println(input + " is a valid student ID.");
  }
}
```

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

```
e018-1234 # Your input
Contains an invalid digit. # Output
```

Console

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

Student ID Validator 1 (Whole code)

```java
import java.util.Scanner;

// Validate input student ID (20XX-XXXXX)
public class StudentIDValidator {

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String input = scanner.next();

    // Validate whether the input string length is 10
    if (input.length() != 10) {
      System.out.println("The input length should be 10.");
      return;

      // Validate whether the 5th character of the input is '-'
    } else if (input.charAt(4) != '-') {
      System.out.println("Fifth character should be '-'.");
      return;

      // Validate whether the characters of the input except 5th character is digits (0, 1, ..., 9)
    } else {
      for (int i = 0; i < input.length(); i++) {
        if (i != 4) {
          char ch = input.charAt(i);
          if (ch < '0' || ch > '9') {
            System.out.println("Contains an invalid an invalid digit.");
            return;
          }
        }
      }
    }

    System.out.println(input + " is a valid student ID.");
  }
}
```

Student ID Validator 2 - 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`.

Student ID Validator 2 - Refactoring

```
import java.util.Scanner;

// Validate input student ID (20XX-XXXXX)
public class StudentIDValidator {

  // Validate whether the input string length is 10
  static boolean isProperLength(String input) {
    return input.length() == 10;
  }

 // Validate whether the 5th character of the input is '-'
  static boolean hasProperDivision(String input) {
    return input.charAt(4) == '-';
  }
...
```

Student ID Validator 2 - Refactoring

```
...
  // Validate whether the characters of the input except
5th character is digits (0, 1, ..., 9)
  static boolean hasProperDigits(String input) {
    for (int i = 0; i < input.length(); i++) {
      if (i != 4) { // 5th character should be '-'
        char ch = input.charAt(i);
        // Check
        if (ch < '0' || ch > '9') {
          return false;
        }
      }
    }
    // All characters but 5th are digit and return true
    return true;
  }
...
```

Student ID Validator 2 - Refactoring

```
...
  // Validate the input student ID and print the
corresponding message
  static void validateStudentID(String input) {
    if (!isProperLength(input)) {
      System.out.println("The input length should be 10.");
    } else if (!hasProperDivision(input)) {
      System.out.println("Fifth character should be '-'.");
    } else if (!hasProperDigits(input)) {
      System.out.println("Contains an invalid digit.");
    } else {
      System.out.println(input + " is a valid student
ID.");
    }
  }
...
```

Student ID Validator 2 - Refactoring

```
...
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String input = scanner.next();
    validateStudentID(input);
  }
}
```

- Check whether the refactored code is running well

Student ID Validator 2 (Whole code)

```java
import java.util.Scanner;

// Validate input student ID (20XX-XXXXX)
public class StudentIDValidator2 {

  // Validate whether the input string length is 10
  static boolean isProperLength(String input) {
    return input.length() == 10;
  }

  // Validate whether the 5th character of the input is '-'
  static boolean hasProperDivision(String input) {
    return input.charAt(4) == '-';
  }

  // Validate whether the characters of the input except 5th character
is digits (0, 1, ..., 9)
  static boolean hasProperDigits(String input) {
    for (int i = 0; i < input.length(); i++) {
      if (i != 4) { // 5th character should be '-'
        char ch = input.charAt(i);
        // Check
        if (ch < '0' || ch > '9') {
          return false;
        }
      }
    }
    // All characters but 5th are digit and return true
    return true;
  }
...
```

```java
...
  // Validate the input student ID and print the corresponding message
  static void validateStudentID(String input) {
    if (!isProperLength(input)) {
      System.out.println("The input length should be 10.");
    } else if (!hasProperDivision(input)) {
      System.out.println("Fifth character should be '-'.");
    } else if (!hasProperDigits(input)) {
      System.out.println("Contains an invalid digit.");
    } else {
      System.out.println(input + " is a valid student ID.");
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String input = scanner.next();
    validateStudentID(input);
  }
}
```

Student ID Validator 3 - Repeated Input

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

Student ID Validator 3 - Repeated Input

- Change yellow highlighted lines

```
...
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String input = scanner.next(); // Get the first input

    // You should compare string with ".equals" in Java!
    while (!input.equals("exit")) {
      validateStudentID(input);

      // Get new input string from the console
      input = scanner.next();
    }
  }
}
```

Add this part

Student ID Validator 3 (Whole Code)

```java
import java.util.Scanner;

// Validate input student ID (20XX-XXXXX)
public class StudentIDValidator3 {

  // Validate whether the input string length is 10
  static boolean isProperLength(String input) {
    return input.length() == 10;
  }

  // Validate whether the 5th character of the input is
'-'
  static boolean hasProperDivision(String input) {
    return input.charAt(4) == '-';
  }

  // Validate whether the characters of the input
except 5th character is digits (0, 1, ..., 9)
  static boolean hasProperDigits(String input) {
    for (int i = 0; i < input.length(); i++) {
      if (i != 4) { // 5th character should be '-'
        char ch = input.charAt(i);
        // Check
        if (ch < '0' || ch > '9') {
          return false;
        }
      }
    }
    // All characters but 5th are digit and return true
    return true;
  }
...
```

```java
...
  // Validate the input student ID and print the
corresponding message
  static void validateStudentID(String input) {
    if (!isProperLength(input)) {
      System.out.println("The input length should be
10.");
    } else if (!hasProperDivision(input)) {
      System.out.println("Fifth character should be
'-'.");
    } else if (!hasProperDigits(input)) {
      System.out.println("Contains an invalid digit.");
    } else {
      System.out.println(input + " is a valid student
ID.");
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Get the first input
    String input = scanner.next();

    // You should compare string with ".equals" in
Java!
    while (!input.equals("exit")) {
      validateStudentID(input);

      // Get new input string from the console
      input = scanner.next();
    }
  }
}
```

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	ee1812345
Output	Contains an invalid digit.
Input	2018-12345
Output	2018-12345 is a valid student ID.

Have a Great Holiday!