CSE3040 Java Language
Exercises

Dept. of Computer Engineering,
Sogang University

This material may have copyright issues. Do not post it on the Internet.

Java Exercises

- This is a Do-It-Yourself material, and will not be covered in lectures.
  - This PPT slides will be constantly updated, so check for updates!

- Each exercise will have a set of requirements that you need to implement, and one or multiple running examples to show you how the program should run when implemented.

- A subset of the exercises will be assigned as homework, and you need to submit your source code for those problems.

- When you write a code, remember to properly document your code.
  - Assume you are posting Java lessons on your blog.
  - Documentation will be an important part of the evaluation.

Level 1

- Requirement(s)
  - Write a Java program that prints “Hello, Java!” on screen, and prints your name on the next line.

- Example output

```
Hello, Java
Jungmin So
```

Level 2

- Requirement(s)
  - Write a Java program that prints “What is your name?” on screen, and then waits for user input. When the user types in the *name*, the program prints “Hello, *name*!” on the screen.

- Example output

```
What is your name?
Jungmin
Hello, Jungmin!
```

Blue: user input

- Hint
  - Use the **Scanner** class
  - https://www.w3schools.com/java/java_user_input.asp

Level 3

- Requirement(s)
  - Write a Java program that asks user for two numbers, and prints the sum of the two numbers.
  - The user may input integers or floating-point numbers. The numbers could also be negative.
  - You don't need to consider erroneous inputs such as non-numbers. Assume the user correctly inputs a single number for each question.

- Example output

```
Enter first number: 3
Enter second number: 2.5
The sum of two numbers is: 5.5
```

- Hint
  - Check out the difference between **System.out.println** and **System.out.print**.
  - Remember you can convert a string into a number and vice versa.

Level 4

- Requirement(s)
  - Write a Java program that asks user for an alphabet, and prints the ASCII code of the alphabet.
  - Assume that the user will only enter a single uppercase letter (A-Z), or a single lowercase letter (a-z). Other inputs such as multiple letters or characters other than alphabets may lead to unexpected outputs.

- Example output

```
ASCII code teller. Enter a letter: C
The ASCII code of C is 67.
```

- Hint
  - ASCII code look up: https://ko.wikipedia.org/wiki/ASCII
  - The String has a method called **charAt**. Check it out!
    - https://www.w3schools.com/java/ref_string_charat.asp

Level 5

- Requirement(s)
  - Write a Java program that does the following.
    - When the program begins, the program draws random number from 1 to 100.
    - Then, the program asks users to guess a number between 1 and 100.
    - If the user input is smaller than the correct answer, print “Too small!” and ask again.
    - If the user input is larger than the correct answer, print “Too large!” and ask again.
    - If the user input is correct, print “Correct!” and end the program.
    - Assume the user will always enter a number between 1 and 100.

- Example output

```
Guess a number (1-100): 37
Too small!
Guess a number (1-100): 53
Too large!
Guess a number (1-100): 44
Correct!
```

Level 5+

- Additional Requirement(s)
  - Count how many times the user inputs an answer.
  - After each wrong answer, update the range of numbers the user should guess from in the next prompt.

- Example output

```
[1] Guess a number (1-100): 37
Too small!
[2] Guess a number (38-100): 53
Too large!
[3] Guess a number (38-52): 44
Correct!
```

Level 6

- Requirement(s)
  - When the program begins, the program draws six numbers from 1 to 45.
    - Careful, all six numbers must be different!
  - Ask the user to input six numbers, one by one.
  - Print how many numbers of user input match the program-selected numbers.
  - Assume that the user will correctly enter different six numbers in the range.
- Example output

```
[Lotto] Enter number #1 (1-45): 2
[Lotto] Enter number #2 (1-45): 33
[Lotto] Enter number #3 (1-45): 27
[Lotto] Enter number #4 (1-45): 44
[Lotto] Enter number #5 (1-45): 10
[Lotto] Enter number #6 (1-45): 7
This week's lotto numbers: 3 10 25 33 34 40
You matched 2 numbers.
```

- Hint
  - To generate a random number, you can use **Math.random()**, or **java.util.Random**.
    - Just use them well to generate an integer between the range we want.

Level 7

- Requirement(s)
  - Write a Java program that counts instances of a letter in a text.
  - When the program starts, the program asks user for a text.
  - Once the user inputs the text, then the program asks users for a letter.
  - The program prints how many instances of the given letter are there in the text.

- Example run(s)

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a letter: n
There are 5 n's in the text.
```

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a letter: z
There are 0 z's in the text.
```

Level 8

- Requirement(s)
  - Write a Java program that counts instances of a string in a text.
  - When the program starts, the program asks user for a text.
  - Once the user inputs the text, then the program asks users for a string
  - The program prints how many instances of the given letter are there in the text.

- Example run(s)

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a string: am
I have found 2 instances of “am”.
```

```
Enter a text: Hello, my name is John. I am an undergraduate student.
Enter a string: that
I have found 0 instances of “that”.
```

- Hint
  - Use methods of class String.

Level 9

- Requirement(s)
  - Write a Java program that finds three students with the best exam scores.
  - When the program starts, the program asks users for exam scores of five students.
  - Then, the program prints the 1st, 2nd, and 3rd place students and their scores.
  - You can assume that there is no two students with the same score.

- Example run(s)

```
Please enter exam scores of each student.
Score of student 1: 50
Score of student 2: 70
Score of student 3: 30
Score of student 4: 90
Score of student 5: 40
The 1st place is student 4 with 90 points.
The 2nd place is student 2 with 70 points.
The 3rd place is student 1 with 50 points.
```

Level 10

- Requirement(s)
  - Write a Java program that multiplies two matrices.
  - The matrices are given as constants variables.
    - Insert the following code at the beginning of the main method.

```
final int A[][] = { {1,2}, {3,4}, {5,6} };
final int B[][] = { {1,2,3,4}, {5,6,7,8} };
```

  - Your program should print the multiplication of the two matrices.
    - Your program should work without modification, even when A and B are changed.
    - It is assumed that A x B is feasible. That is, if A is a **m**x**n** matrix, B is a **n**x**p** matrix.

- Example run(s)

```
A
1 2
3 4
5 6

B
1 2 3 4
5 6 7 8

AxB
11 14 17 20
23 30 37 44
35 46 57 68
```

Level 11

- Goal(s)
  - We want to print the Fibonacci sequence. The definition of Fibonacci sequence is as follows.
    - f(0) = 0, f(1) = 1, f(n) = f(n-1) + f(n-2)
  - We have implemented a part of the code. Complete the missing part so that we get the output when we run the program.
- Requirements
  - Do not touch class Level011.
  - You must not use while loop or for loop.
  - IntSequence should be defined as an interface.
- Implemented part of the code

```
public class Level011 {
    public static void main(String[] args) {
        IntSequence seq = new FibonacciSequence();
        for(int i=0; i<20; i++) {
            if(seq.hasNext() == false) break;
            System.out.print(seq.next() + " ");
        }
        System.out.println(" ");
    }
}
```

- Expected output

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
```

Level 12

- Goal(s)
  - We want to write a program that takes an integer from user and prints the integer in binary number format.
  - We have implemented a part of the code. Complete the missing part so that we get the output when we run the program.
- Requirements
  - Do not touch class Level012.
  - IntSequenceStr should be defined as an interface.
- Tip: If necessary, you may use Math.pow(). E.g.) Math.pow(2, 3) returns 2^3 which is 8.
- Implemented part of the code

```
public class Level012 {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a positive integer: ");
        String str = in.nextLine();
        int num = Integer.parseInt(str);
        in.close();
        System.out.println("Integer: " + num);
        IntSequenceStr seq = new BinarySequenceStr(num);
        System.out.print("Binary number: ");
        while(seq.hasNext()) System.out.print(seq.next());
        System.out.println("");
    }
}
```

- Example run

```
Enter a positive integer: 64
Integer: 64
Binary number: 1000000
```

Level 13

- Goal(s)
  - We want to write a program that calculates the sum of area of multiple shapes.
  - Supported shapes are circles, squares and rectangles.
  - We have implemented a part of the code. Complete the missing part so that we get the output when we run the program.
- Requirement(s)
  - Do not touch the main method in class Level013.
  - If you define classes, all instance variables must be defined as private variables.
- Tip(s)
  - You can use the constant Math.PI for calculating area of circles.

- Implemented part of the code

```
public class Level013 {
    public static void main(String args[]) {
        Shape[] arr = {new Circle(5.0), new Square(4.0), new Rectangle(3.0, 4.0), new Square(5.0)};
        System.out.println("Total area of the shapes is: " + sumArea(arr));
    }
}
```

- Output

```
Total area of the shapes is: 131.53981633974485
```

Level 14

- Goal(s)
  - We want to write a program that calculates distance between two points in N-dimensional space.
  - The program supports two distance metrics: EuclideanDistance and ManhattanDistance.
    - Euclidean distance between two N-dimensional points (p_1, p_2, ..., p_n), (q_1, q_2, ..., q_n) is:
      $d = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ ... + (p_n - q_n)^2}$
    - Manhattan distance between two N-dimensional points (p_1, p_2, ..., p_n), (q_1, q_2, ..., q_n) is:
      $d = |\ p_1\ - q_1\ | + |\ p_2 - q_2| + \cdots + |\ p_n - q_n|$
  - When the compared two points have different dimensions, then the printed distance is -1.
  - We have implemented a part of the code. Complete the missing part so that we get the output when we run the program.
- Requirements
  - Do not touch class Level014.

- Implemented part of the code

```
public class Level014 {
    public static void main(String[] args) {
        Point p1 = new Point(new double[] {1.0, 2.0, 3.0});
        Point p2 = new Point(new double[] {4.0, 5.0, 6.0});
        System.out.println("Euclidean Distance: " + EuclideanDistance.getDist(p1, p2));
        System.out.println("Manhattan Distance: " + ManhattanDistance.getDist(p1, p2));
        Point p3 = new Point(new double[] {1.0, 2.0, 3.0});
        Point p4 = new Point(new double[] {4.0, 5.0});
        System.out.println("Euclidean Distance: " + EuclideanDistance.getDist(p3, p4));
        System.out.println("Manhattan Distance: " + ManhattanDistance.getDist(p3, p4));
    }
}
```

Level 15

- Goal(s)
  - We want to write a program that compares two instances of class Points.
  - Class Points has an array of floating-point numbers.
  - Two Points objects are regarded equal if the sum of floating-point numbers are equal.
  - Also, if the user prints a Points objects, it should be printed as shown in the Result.
  - class Level015 is provided. You must complete the program to get the output as below.

- Requirements
  - Do not touch class Level015.
  - You do not need to implement method hashCode.

Level 15 (cont.)

- Implemented part of the code

```
public class Level015 {
    public static void main(String[] args) {
        Points p1 = new Points(new double[] {1.0, 2.0, 3.0});
        Points p2 = new Points(new double[] {4.0, 5.0, 6.0});
        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p1.equals(p2));
        Points p3 = new Points(new double[] {1.0, 4.0, 7.0});
        Points p4 = new Points(new double[] {3.0, 9.0});
        System.out.println(p3);
        System.out.println(p4);
        System.out.println(p3.equals(p4));
        Points p5 = new Points(new double[] {1.0, 2.0});
        Points p6 = null;
        System.out.println(p5);
        System.out.println(p6);
        System.out.println(p5.equals(p6));
    }
}
```

- Result

```
[sum of points: 6.0]
[sum of points: 15.0]
false
[sum of points: 12.0]
[sum of points: 12.0]
true
[sum of points: 3.0]
null
false
```

Level 16

- Goal(s)
  - We want to write a program that reads text from a file and counts number of occurrences for each character.
  - The file contains multiple lines of text. For example:

```
Java is a general-purpose programming language that is class-based, object-oriented (although not a pure object-oriented language, as it contains primitive types), and designed to have as few implementation dependencies as possible.
It is intended to let application developers write once, run anywhere (WORA) meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.
Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.
As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.
```

  - The program should output results as shown in the next slide.
  - If the file does not exist, the program should print out an error message. The program must not crash.
  - class Level016 is provided. Complete the program so that it will run as the example below.

- Requirements
  - Do not touch class Level016. You may only change the file name.

Level 16 (cont.)

- Implemented part of the code
  - You may only change the part colored in blue.

```
public class Level016 {
    public static void main(String args[]) {
        Text t = new Text();
        t.readTextFromFile("src/cse3040/exercises/input_Level016.txt");
        System.out.println("a: " + t.countChar('a'));
        System.out.println("b: " + t.countChar('b'));
        System.out.println("c: " + t.countChar('c'));
    }
}
```

- Example Run
  - We assume that the input file contained text from the previous slide.

```
a: 71
b: 8
c: 26
```

  - If the input file does not exist:

```
Error: file does not exist.
a: 0
b: 0
c: 0
```

Level 17

- Goal(s)
  - We want to write a program that reads data from file and prints out a summary of data.
  - The file contains items and their prices. Each line contains an item and its price, separated by a space.
  - An example input file is:

```
apple 3.99
watermelon 11.99
strawberries 4.99
pear 6.99
mango 8.99
bananas 4.99
pineapples 7.99
kiwi 5.99
raspberries 2.99
peaches 5.99
```

  - The program should read data from the file and write its summary to a file. Assuming the input file contains the above data, your output file should look like this:

```
Summary
-------
number of items: 10
most expensive item: watermelon (11.99)
cheapest item: raspberries (2.99)
average price of items: 6.49
```

Level 17

- Requirements(s) - read carefully and follow the instructions.
  - For this problem, there is no given code. You should write the program from scratch.
  - Your public class name must be Level017.
  - Content of the input file could be changed. Still, your program should produce correct results based on the input data.
  - The name of the input file is fixed to "./input.txt". The input file should be in the project base directory.
  - If the file does not exist, your program should print "Input file not found!" and terminate normally. Your program must not crash! (crash means the console displays exceptions and stack trace.)
  - Other than the "file not found" problem, you can ignore all other exceptions. You may use try-catch blocks, or you may throw these exceptions.
  - If the input file exists, you can assume that the file content is in the correct format. You do not need to implement error handling routine for wrong input formats.
  - In the list of items, you can assume that all items have different names. Also, an item name can only contain lowercase alphabet letters. (No uppercase letters or special characters are used.)
  - Multiple items could have the same price. If there are multiple most expensive items, you can choose any one of them as the most expensive item. Similarly, if there are multiple cheapest items, you can choose any one of them as the cheapest item.
  - The price is a non-negative real value.
  - You are free to use any library provided by Java. However, do not use external libraries.

Level 18

- Goal(s)
  - We want to write a program that reads data from file and prints out the data in a sorted order.
  - An example input file is:

```
apple 3.99
watermelon 11.99
strawberries 4.99
pear 6.99
mango 8.99
bananas 4.99
pineapples 7.99
kiwi 5.99
raspberries 2.99
peaches 5.99
```

  - Then, your program must sort the items in the ascending order of price. If multiple items have the same price, they should be sorted in the alphabetical order of the item names.
    - "bananas 4.99" appears before "strawberries 4.99".
  - The output should be:

```
raspberries 2.99
apple 3.99
bananas 4.99
strawberries 4.99
kiwi 5.99
peaches 5.99
pear 6.99
pineapples 7.99
mango 8.99
watermelon 11.99
```

Level 18

- Requirement(s)
  - We have implemented a part of the code. (shown below)
  - You must not modify the main method of class Level018. Other than that, you are free to insert your code into the file such as defining new classes, methods, and variables inside or outside class Level018.
  - Similar to Level 17, if the input file does not exist, your program should print "input file not found!" and terminate normally. Other than that, you do not need to handle any exceptions.
  - If the input file exists, the assumptions on the input file are the same as Level 17. The content of the file is in correct format, all item names are in lowercase alphabet letters, all item names are unique, and multiple items may have the same price.

```
/* insert your code here! */

public class Level018 {

    /* insert your code here! */

    public static void main(String[] args) throws Exception {
        ArrayList<Element> list = new ArrayList<>();
        int rv = readDataFromFile("./input.txt", list);
        if(rv == 1) {
            System.out.println("input file not found!");
            return;
        }
        Collections.sort(list);
        Iterator<Element> it = list.iterator();
        while(it.hasNext()) System.out.println(it.next());
    }
}
```

Level 19

- Goal(s)
  - Similar to Level 18, We want to write a program that reads data from file and prints out the data in a sorted order. This time, we will use a map instead of a list.
  - The file contains items and their prices. Each line contains an item and its price, separated by a space.
  - An example input file is:

```
apple 3.99
watermelon 11.99
strawberries 4.99
pear 6.99
mango 8.99
bananas 4.99
pineapples 7.99
kiwi 5.99
raspberries 2.99
peaches 5.99
```

  - Your program should produce the data, sorted by the item name in alphabetical order.
    - Note that this output is different from the output of Level 018.

```
apple 3.99
bananas 4.99
kiwi 5.99
mango 8.99
peaches 5.99
pear 6.99
pineapples 7.99
raspberries 2.99
strawberries 4.99
watermelon 11.99
```

Level 19

- Requirement(s)
  - We have implemented a part of the code. (shown below)
  - You must not modify the main method of class Level019. Other than that, you are free to insert your code into the file such as defining new classes, methods, and variables inside or outside class Level019.
  - Similar to Level 17 and 18, if the input file does not exist, your program should print "input file not found!" and terminate normally. Other than that, you do not need to handle any exceptions.
  - If the input file exists, the assumptions on the input file are the same as Level 17 and 18. The content of the file is in correct format, all item names are in lowercase alphabet letters, all item names are unique, and multiple items may have the same price.

```
/* insert your code here! */

public class Level019 {

    /* insert your code here! */

    public static void main(String[] args) throws Exception {
        Map<String,Double> map = InitializeMap();
        int rv = readDataFromFile("./input.txt", map);
        if(rv == 1) {
            System.out.println("input file not found!");
            return;
        }
        System.out.println(map);
    }
}
```

Level 20

- Goal(s)
  - Similar to Level 19, we are going to use a map to print out the data in a sorted order.
  - This time, the output should be the same as Level 18. The items must be sorted in the ascending order of price. If multiple items have the same price, they should be sorted in the alphabetical order of the item names.
  - The output should be as follows. (The input is similar to the input in Levels 17-19.)

```
raspberries 2.99
apple 3.99
bananas 4.99
strawberries 4.99
kiwi 5.99
peaches 5.99
pear 6.99
pineapples 7.99
mango 8.99
watermelon 11.99
```

- Requirement(s)
  - The requirements are the same as Level 19, except that your public class name is changed to Level020.
  - You should use the same main method that is shown in Level 19.

Level 21

- Goal(s)
  - We want to build a program that fetches a URL on the Internet and finds the information we need.
  - Specifically, we want to find the 50 weekly bestseller books (주간 종합 베스트셀러) from the website "알라딘", which is a Korean online bookstore.
  - You can find the webpage at:
    - https://aladin.co.kr/ shop/common/wbest.aspx?BestType=Bestseller&BranchType=1&CID=0
  - From this site, you should fetch the 50 bestseller books, and print them on the screen like the example shown in the next slide.
    - The first line should indicate the date (year, month, week).
    - From the second line, the title and the author of each bestseller book should be displayed in the order of rank.
    - Your output should follow the format shown in the next slide.

- Requirement(s)
  - Your public class should be named Level021.
  - For this problem, you may not use any external libraries.
  - "2019년 11월 4주" must also be fetched from the URL, because it shows the current week. Once the web page updates the date to "2019년 12월 1주", your program should show the current week without modifying the code.

Level 21

- Example output

```
[2019년 11월 4주]
1위: 나의 마녀 2 (해윤)
2위: 1일 1페이지, 세상에서 가장 짧은 교양 수업 365 (데이비드 S. 키더)
3위: 설민석의 한국사 대모험 12 (설민석)
4위: 트렌드 코리아 2020 (김난도)
5위: 에이트 (이지성)
6위: Go Go 카카오프렌즈 11 : 한국 (김미영)
7위: 지금 이대로 좋다 (법륜)
8위: 넛지 (리처드 H. 탈러)
9위: 심신 단련 (이슬아)
10위: 여행의 이유 (김영하)
11위: 나를 돌보지 않는 나에게 (정여울)
12위: 작가들의 비밀스러운 삶 (기욤 뮈소)
13위: 일의 기쁨과 슬픔 (장류진)
14위: 날씨의 아이 공식 비주얼 가이드 (신카이 마코토)
15위: 호텔 창문 (편혜영)
16위: 82년생 김지영 (조남주)
17위: 오래도록 젊음을 유지하고 건강하게 죽는법 (스티븐 R. 건드리)
18위: 공부머리 독서법 (최승필)
19위: 아홉 명의 완벽한 타인들 (리안 모리아티)
20위: 지쳤거나 좋아하는 게 없거나 (글배우)
21위: 2020 박막례 일력 (박막례)
22위: 5년 후 나에게 - Q&A a day (2020 로즈골드 특별 한정판) (포터 스타일)
23위: 인생은 짧고 고양이는 귀엽지 (이용한)
24위: 방랑자들 (올가 토카르추크)
25위: 2020 현직교사들이 들려주는 면접레시피 (류은진)
26위: 고미숙의 글쓰기 특강 : 읽고 쓴다는 것, 그 거룩함과 통쾌함에 대하여 (고미숙)
27위: 식물의 책 (이소영)
28위: 해커스 토익 기출 보카 TOEIC VOCA 단어장 (David Cho)
29위: 총몽 Last Order 라스트 오더 완전판 3 (기시로 유키토)
30위: 해리 포터와 마법사의 돌 (일러스트 에디션) (J.K. 롤링)
31위: 총몽 Last Order 라스트 오더 완전판 4 (기시로 유키토)
32위: 90년생이 온다 (리커버 특별판) (임홍택)
33위: 부의 인문학 (브라운스톤)
34위: 사피엔스 (유발 하라리)
35위: 아주 작은 습관의 힘 (제임스 클리어)
36위: 총 균 쇠 (반양장) (재레드 다이아몬드)
37위: 선량한 차별주의자 (김지혜)
38위: 흔한남매 2 (흔한남매)
39위: 튤립 사운드북 크리스마스 캐럴 2019 (민유경)
40위: 도어 (서보 머그더)
41위: 혼자가 혼자에게 (이병률)
42위: 2020 선재국어 한 권으로 정리하는 마무리 (이선재)
43위: 흔한남매 1 (흔한남매)
44위: 란마 1/2 애장판 3 (다카하시 루미코)
45위: 깨끗한 존경 (이슬아)
46위: ETS 토익 정기시험 기출문제집 1000 Reading (ETS)
47위: 그 사랑 놓치지 마라 (이해인)
48위: 메종일각 신장판 3 (다카하시 루미코)
49위: 메종일각 신장판 4 (다카하시 루미코)
50위: 이상한 과자 가게 전천당 3 (히로시마 레이코)
```

Level 22

- Goal(s)
  - The goal of Level 22 is the same as Level 21, except that this time, you should use jsoup library to achieve the goal.
  - We want to build a program that fetches a URL on the Internet and finds the information we need.
  - Specifically, we want to find the 50 weekly bestseller books (주간 종합 베스트셀러) from the website "알라딘", which is a Korean online bookstore.
  - You can find the webpage at:
    - https://aladin.co.kr/ shop/common/wbest.aspx?BestType=Bestseller&BranchType=1&CID=0
  - From this site, you should fetch the 50 bestseller books, and print them on the screen like the example shown in the previous slide.
    - The first line should indicate the date (year, month, week).
    - From the second line, the title and the author of each bestseller book should be displayed in the order of rank.
    - Your output should follow the format shown in the previous slide.

- Requirement(s)
  - Your public class should be named Level022.
  - For this problem, you should use jsoup library to achieve the goal.
  - "2019년 11월 4주" must also be fetched from the URL, because it shows the current week. Once the web page updates the date to "2019년 12월 1주", your program should show the current week without modifying the code.

End of Slides

Instructor office: AS818A

Email: jso1@sogang.ac.kr