**Computer Language 2022**

**Assignment #3**

**Due: 11/Apr 23:59:59**

**Student ID: 21102052 Name: Lee Jeong-Yun**

**For each question, 1) write your solution codes, 2) present a screenshot of your result, and 3) describe a short explanation about your solution. Without these components, you will be given some penalties.**

1. **Write a program to take a positive number from the user and print out the reversed one.   
   You cannot use String/Collection classes’ reverse-related functions.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Output)**

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

**Your code:**

import java.util.Scanner;  
public class Assignment03\_01 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Input your number: ");  
 long num = sc.nextLong();  
 long result = 0l;  
 while (num != 0) {  
 result = result \* 10 + num % 10;  
 num = num / 10;  
 }  
 System.*out*.print("Result: " + result);  
 }  
}

**Your result (screenshot):**

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

**Your explanation:**

By using scanner, get integer from the user.

To correspond to the big input value and big result value, I used long type variables for input and result.

Since the conditions to ban using string-related function, I introduced the while loop statement to get each digit from the input value from one’s digit to end of digits. (e.g., input: 13245 -> then get 5, 4, 2, 3, 1 in descending order) When the program get single digit from the input, result value is multiplied by 10 and the digit is added to result variable and the result value.

After assigning the single digit, then input value is divided by 10 to remove the used digit and get next digit. If the input value is not remained, then the loop is ended and print out the result.

1. **Implement Grade class to make the following program work. Grade class’s main method is as follows:**

public static void main(String [] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Input the score of Math, Science, and English subject >> ");  
 int math = scanner.nextInt();  
 int science = scanner.nextInt();  
 int english = scanner.nextInt();

Grade me = new Grade(math, science, english);

System.*out*.println("Average is " + me.average());   
 scanner.close();  
}

**Output)**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Your code:**

import java.util.Scanner;  
  
public class Grade {  
 public int math;  
 public int science;  
 public int english;  
 public Grade(int math, int science, int english){  
 this.math = math;  
 this.science = science;  
 this.english = english;  
 }  
 public int average(){  
 int avg;  
 avg = (math+science+english)/3;  
 return avg;  
 }  
  
 public static void main(String [] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Input the score of Math, Science, and English subject >> ");  
 int math = scanner.nextInt();  
 int science = scanner.nextInt();  
 int english = scanner.nextInt();  
  
 Grade me = new Grade(math, science, english);  
  
 System.out.println("Average is " + me.average());  
 scanner.close();  
 }  
}

**Your result (screenshot):**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**Your explanation:**

What I should write is Grade class for running given main method.

I made public class Grade and added public field for three variables, math, science, and english.

And then, write the constructor of Grade class to get parameters by using the input in ( ), brackets.

In constructor phrase, using ‘this’ keyword to assign the input values to class field.

Also, main method requires to make average method to calculate and return the average of three variables.

Then, we can find out the main method works successfully.

1. **Implement Rectangle class to provide the following features:**

* Member fields: x, y, width, height (int type)
* Constructor(…): initializes x, y, width, and height fields using the given arguments
* getArea(): returns the area of the rectangle
* show(): prints the coordinate and the area of the rectangle
* contains(Rectangle rect): returns *true* if the current object contains the given Rectangle object *rect*
* The main method of Rectangle class and the expected output is as follows:

텍스트이(가) 표시된 사진

자동 생성된 설명

**Conceptual diagram)**

어두운이(가) 표시된 사진

자동 생성된 설명

**Output)**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Your code:**

public class Rectangle {  
 public int x;  
 public int y;  
 public int width;  
 public int height;  
  
 public Rectangle(int x,int y, int width, int height){  
 this.x = x;  
 this.y = y;  
 this.width = width;  
 this.height = height;  
 }  
  
 public int getArea(){  
 int area;  
 area = width\*height;  
 return area;  
 }  
  
 public void show(){  
 System.out.print("coordinate: ("+x+","+y+")");  
 System.out.println(" area: "+width+"x"+height);  
 }  
  
 boolean tf;  
  
 public boolean contains(Rectangle rect){  
 tf = false;  
 if (rect.x >= this.x) {  
 if (rect.x+rect.width <= this.x+this.width){  
 if (rect.y >= this.y){  
 if (rect.y+rect.height <= this.y+this.height){  
 tf = true;  
 }  
 }  
 }  
 }  
  
 return tf;  
 }  
  
 public static void main(String args[]){  
 Rectangle r = new Rectangle(2,2,9,7);  
 Rectangle s = new Rectangle(5,5,7,7);  
 Rectangle t = new Rectangle(1,1,10,10);  
  
 r.show();  
 System.out.println("s' area is "+s.getArea());  
 if (t.contains(r)){  
 System.out.println("t contains r.");  
 }  
 if (t.contains(s)){  
 System.out.println("t contains s.");  
 }  
 }  
}

**Your result (screenshot):**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**Your explanation:**

Rectangle should get 4 variables by bracket. So, I made public fields with 4 variables and wrote Rectangle constructor to get parameters. I added 4 variables in bracket, and they are assigned to public field by ‘this’ keyword.

In given main method, it asked to make 3 method, getArea(), show(), and contains(Rectangle rect)

getArea() -> make integer variable, area and width\*height is assigned to it. this method returns area variables.

show() -> print out coordinates from parameters and area from getArea() method.

contains(Rectangle rect) -> before the write the inner statement in the method, I made Boolean variable, tf. It will take a role of indicator to check whether each case is satisfied by conditions. Default value of tf is false. If rect is included in the instance Rectangle, then our indicator will be changed to true. rect’s coordinates should be smaller than current coordinates and the rect’s x+width and y+height also should be smaller. These conditions mean current object contains the given object.

1. **Write a program that works as follows. Your program must have Phone class (private name, tel fields and their getters, and constructor should be implemented), PhoneBook class. PhoneBook class must take the number of people to store from the user and creates Phone array. Information of a single person is stored in a single Phone instance. If a user types "stop", the program ends.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Your code:**

import java.util.Scanner;  
  
class Phone {  
 private String name;  
 private String tel;  
  
 public String getName() {  
 return name;  
 }  
 public String getTel() {  
 return tel;  
 }  
  
 public Phone(String name, String tel) {  
 this.name = name;  
 this.tel = tel;  
 }  
}  
  
  
public class PhoneBook {  
  
 Phone[] PhoneArray;  
  
 private Scanner scanner;  
  
 public PhoneBook() {  
 scanner = new Scanner(System.*in*);  
 }  
  
 void read() {  
 System.*out*.print("Number of person to store? >> ");  
 // implement here to take user info and store them into the array  
 int numberOfPeople = scanner.nextInt();  
 PhoneArray = new Phone[numberOfPeople];  
 for (int i=0; i<numberOfPeople; i++){  
 System.*out*.print("Name and Tel no. >> ");  
 String inputName = scanner.next();  
 String inputTel = scanner.next();  
 Phone record = new Phone(inputName,inputTel);  
 PhoneArray[i] = record;  
 }  
 System.*out*.println("Saved.");  
 }  
  
 boolean success;  
 String search(String name) {  
 String tel = "";  
 for (int j=0;j<PhoneArray.length;j++) {  
 success = false;  
 if (name.equals(PhoneArray[j].getName())){  
 tel = PhoneArray[j].getTel();  
 success = true;  
 break;  
 }  
 }  
 return tel;  
 }  
  
 void run() {  
 read();  
 while(true) {  
 System.*out*.print("Who do you wanna search for? >> ");  
 String findName = scanner.next();  
 String seek = search(findName);  
 if (findName.equals("stop")){  
 break;  
 }  
 else if (success==true){  
 System.*out*.print(findName+"'s telephone no. >> ");  
 System.*out*.println(seek);  
 }  
 else if (success==false){  
 System.*out*.println(findName+" does not exist.");  
 }  
 }  
 scanner.close();  
 }  
  
 public static void main(String[] args) {  
 new PhoneBook().run();  
 }  
}

**Your result (screenshot)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**Your explanation:**

Phone class (private name, tel fields and their getters, and constructor should be implemented)

* Private field with name and tel is added. getName() and getTel() is added as getter because name and tel is private. Constructor of Phone is added to assign input value to name and tel private variables.

PhoneBook class must take the number of people to store from the user and creates Phone array.

* Array, ‘PhoneArray’ from Phone class is added to PhoneBook class.

read()

* Using Scanner to get number of people user want to store and initializing size of PhoneArray by numberOfPeople. Information of a single person is stored in a single Phone instance and assigned in PhoneArray by for loop statement. In for loop, program asks user to input information of people nth times (n=numberOfPeople). When for loop get some information of single person, the input value is divided into inputName and inputTel and stored in Phone instance, record. The record is assigned each PhoneArrary element. number of element is already decided by numberOfPeople.

search(String name)

* ‘success’ variable is set in Boolean type. ‘success’ takes role of indicator of success or not of searching method.

String tel is initialized in “”. Using for loop, initialize success to false (default) and find the name in search method’s bracket from each PhoneArray. If for loop find the name in the array, then tel variable get specific tel from PhoneArray and success is changed to true and exit the loop. After that, return tel.

run()

* Using Scanner, program get name as findName variable which is user want to find its tel. If a user types "stop", the program ends using ‘break;’. seek variables are added for get tel. If the findName is existing in PhoneArray through search(findName), success would be true. Then, print out “findName’s telephone no. >>” and tel. If search method cannot find the findName in PhoneArray, then success is false and print out “findName does not exist.”

**Template code:**

class Phone {  
  
 public Phone(String name, String tel) {  
  
 }  
  
}  
  
public class PhoneBook {  
 private Scanner scanner;  
  
 public PhoneBook() {  
 scanner = new Scanner(System.*in*);  
 }  
  
 void read() {  
 System.*out*.print("Number of person to store? >> ");

// implement here to take user info and store them into the array  
  
 System.*out*.println("Saved.");  
 }  
  
 String search(String name) { // method to search for a certain user name  
 // implement here

}  
  
 void run() {  
 read();  
 while(true) {  
 System.*out*.print("Who do you wanna search for? >> ");  
 String name = scanner.next();

// implement here

// if a user types “stop”, then program exits.

// otherwise, your program searches for the text and returns the result

}  
  
 scanner.close();  
 }  
  
 public static void main(String[] args) {  
 new PhoneBook().run();  
 }  
}

1. Read carefully the following code. **Find all the lines (among line number 1 ~ 12) where a compile error can occur** and explain the reason for the error for each problematic line.

**public class** myClass {  
 **int objAttribute**; //Non-Static  
 **static int** *staticAttribute*; //Static  
  
 **void** myMethod() { // Non-Static  
 objAttribute= 10; ( 1 )  
 *staticAttribute* = 20; ( 2 )  
 System.***out***.println(**"print!"**);  
 }  
 **static void** yourMethod() { //Static  
 objAttribute= 30; ( 3 )  
 *staticAttribute* = 40; ( 4 )  
 System.***out***.println(**"static print!"**);  
 }  
 **public static void** main(String[] args) {  
 myClass mc = **new** myClass();  
 myClass.myMethod(); ( 5 )  
 myClass.*yourMethod*(); ( 6 )  
 mc.myMethod(); ( 7 )  
 mc.*yourMethod*(); ( 8 )  
 mc.objAttribute= 100; ( 9 )  
 mc.*staticAttribute* = 200; ( 10 )  
 myClass.objAttribute= 300; ( 11 )  
 myClass.*staticAttribute* = 400; ( 12 )  
  
 }  
}

**Your explanation:**

class can use only ‘static’ fields or methods. However, Instance can use fields for methods without static and it also can use static fields or methods. Non-static field or method cannot be referenced from a static context. Class is using only static stuff. Instance is using both static and non-static.

(3): method, ‘yourMethod’ has ‘static’ keyword. This means that the method is available to class and instance. But the variable, ‘objAttribute’ has no that keyword, ‘static’. ‘static’ method requires ‘static’ variable. non-static variables are not accessible for static fields or method.

(5): In ‘myClass.myMethod’, ‘myClass’ calls class, not ask to instantiate. Since the class can only use static fields and methods, ‘myClass’ cannot get ‘myMethod’ which is non-static method. But, as we can find out in (7), if we instantiate the class and ask to call ‘myMethod’, then we can use ‘myMethod’ without any problems.

(11): As with the above problem, this code line is problematic due to the difference between the fields and methods available to the class and the instance. Despite ‘myClass’ is class, it requires to call ‘objAttribute’ which is non-static field.