## CSCI 145 PA \_\_7\_\_ Submission

## Due Date:\_\_\_Apr 17, 2023\_\_\_ Late (date and time):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Name(s):\_\_\_\_\_\_Ivan Leung\_\_\_\_\_\_ & \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Exercise 1 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa7;

//Class Player for CSCI 145 PA 4 Spring 2023

//Modified by: Ivan Leung

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 1 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//ChangingPeople.java

//

//Demonstrates parameter passing -- contains a method that should

//change to Person objects.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** ChangingPeople {

// ---------------------------------------------------------

// Sets up two person objects, one integer, and one String

// object. These are sent to a method that should make

// some changes.

// ---------------------------------------------------------

**public** **static** **void** main(String[] args) {

Person person1 = **new** Person("Sally", 13);

Person person2 = **new** Person("Sam", 15);

**int** age = 21;

String name = "Jill";

System.***out***.println("\nParameter Passing... Original values...");

System.***out***.println("person1: " + person1);

System.***out***.println("person2: " + person2);

System.***out***.println("age: " + age + "\tname: " + name + "\n");

*changePeople*(person1, person2, age, name);

System.***out***.println("\nValues after calling changePeople...");

System.***out***.println("person1: " + person1);

System.***out***.println("person2: " + person2);

System.***out***.println("age: " + age + "\tname: " + name + "\n");

}

// -------------------------------------------------------------------

// Change the first actual parameter to "Jack - Age 101" and change

// the second actual parameter to be a person with the age and

// name given in the third and fourth parameters.

// -------------------------------------------------------------------

**public** **static** **void** changePeople (Person p1, Person p2, **int** age, String name)

{

System.***out***.println ("\nInside changePeople... Original parameters...");

System.***out***.println ("person1: " + p1);

System.***out***.println ("person2: " + p2);

System.***out***.println ("age: " + age + "\tname: " + name + "\n");

//Make changes

// Person p3 = new Person (name, age); // original

// p2 = p3; // original

p2.changeName(name); // new

p2.changeAge(age); //new

name = "Jack";

age = 101;

p1.changeName (name);

p1.changeAge (age);

//Print changes

System.***out***.println ("\nInside changePeople... Changed values...");

System.***out***.println ("person1: " + p1);

System.***out***.println ("person2: " + p2);

System.***out***.println ("age: " + age + "\tname: " + name + "\n");

}

}

**package** pa7;

//Class Player for CSCI 145 PA 4 Spring 2023

//Modified by: Ivan Leung

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Person.java

//

//A simple class representing a person.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** Person {

**private** String name;

**private** **int** age;

// ----------------------------------------------------------

// Sets up a Person object with the given name and age.

// ----------------------------------------------------------

**public** Person(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

// ----------------------------------------------------------

// Changes the name of the Person to the parameter newName.

// ----------------------------------------------------------

**public** **void** changeName(String newName) {

name = newName;

}

// ----------------------------------------------------------

// Changes the age of the Person to the parameter newAge.

// ----------------------------------------------------------

**public** **void** changeAge(**int** newAge) {

age = newAge;

}

// ----------------------------------------------------------

// Returns the person's name and age as a string.

// ----------------------------------------------------------

**public** String toString() {

**return** name + " - Age " + age;

}

}

Input/output below:

Parameter Passing... Original values...

person1: Sally - Age 13

person2: Sam - Age 15

age: 21 name: Jill

Inside changePeople... Original parameters...

person1: Sally - Age 13

person2: Sam - Age 15

age: 21 name: Jill

Inside changePeople... Changed values...

person1: Jack - Age 101

person2: Jill - Age 21

age: 101 name: Jack

Values after calling changePeople...

person1: Jack - Age 101

person2: Jill - Age 21

age: 21 name: Jill

Exercise 2 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa7;

/\* Java Class: CSCI 145

Author: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**public** **class** Compare3 {

**public** **static** Comparable largest(Comparable val1, Comparable val2, Comparable val3) {

Comparable large = val1;

**if** (large.compareTo(val2) < 0) {

large = val2;

}

**if** (large.compareTo(val3) < 0) {

large = val3;

}

**return** large;

}

}

**package** pa7;

/\* Java Class: CSCI 145

Author: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**import** java.util.Scanner;

**public** **class** Comparisons {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str1, str2, str3;

**int** val1, val2, val3;

Scanner scan = **new** Scanner(System.***in***);

System.***out***.print("Enter three strings: ");

str1 = scan.next();

str2 = scan.next();

str3 = scan.next();

System.***out***.println("The largest string: " + Compare3.*largest*(str1, str2, str3) + "\n");

System.***out***.print("Enter three integers: ");

val1 = scan.nextInt();

val2 = scan.nextInt();

val3 = scan.nextInt();

scan.close();

System.***out***.println("The largest integer: " + Compare3.*largest*(val1, val2, val3) + "\n");

}

}

Input/output below:

Enter three strings: abc hello 389

The largest string: hello

Enter three integers: 889 1 32

The largest integer: 889

Exercise 3 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa7;

/\* Java Class: CSCI 145

Author: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**public** **class** RandomWalk {

**private** **int** x;

**private** **int** y;

**private** **int** maxSteps;

**private** **int** currentSteps;

**public** **int** boundary;

**public** RandomWalk(**int** max, **int** edge) {

maxSteps = max;

boundary = edge;

currentSteps = 0;

x = 0;

y = 0;

}

**public** RandomWalk(**int** max, **int** edge, **int** startX, **int** startY) {

maxSteps = max;

boundary = edge;

x = startX;

y = startY;

}

**public** **void** takeStep() {

**int** randomDirection = (**int**) (Math.*random*() \* 3);

**switch** (randomDirection) {

**case** 0:

++x;

++currentSteps;

**break**;

**case** 1:

--x;

++currentSteps;

**break**;

**case** 2:

++y;

++currentSteps;

**break**;

**case** 3:

--y;

++currentSteps;

**break**;

}

}

**public** String toString() {

**return** "Steps: " + currentSteps + "; Position: (" + x + "," + y + ")";

}

**public** **boolean** moreSteps() {

**return** currentSteps < maxSteps;

}

**public** **boolean** inBounds() {

**return** (**int**) Math.*abs*(x) <= boundary && (**int**) Math.*abs*(y) <= boundary;

}

**public** **void** walk() {

**while**(moreSteps() && inBounds()) {

takeStep();

}

}

}

**package** pa7;

/\* Java Class: CSCI 145

Author: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**import** java.util.Scanner;

**public** **class** TestWalk {

**public** **static** **void** main(String[] args) {

RandomWalk walk1 = **new** RandomWalk(10, 5);

RandomWalk walk2;

RandomWalk walk3 = **new** RandomWalk(200, 10);

**int** maxSteps;

**int** boundary;

**int** x;

**int** y;

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Walk 2:");

System.***out***.println("Enter maximum step");

maxSteps = scan.nextInt();

System.***out***.println("Enter boundary");

boundary = scan.nextInt();

System.***out***.println("Enter x");

x = scan.nextInt();

System.***out***.println("Enter y");

y = scan.nextInt();

scan.close();

walk2 = **new** RandomWalk(maxSteps, boundary, x, y);

System.***out***.println("Walk1: " + walk1);

System.***out***.println("Walk2: " + walk2);

System.***out***.println();

**for** (**int** i = 0; i < 5; ++i) {

walk1.takeStep();

walk2.takeStep();

System.***out***.println("Walk 1: " + walk1);

System.***out***.println("Walk 2: " + walk2);

System.***out***.println();

}

System.***out***.println();

System.***out***.println("Walk3 before walk: " + walk3);

walk3.walk();

System.***out***.println("Walk3 after walk: " + walk3);

}

}

Input/output below:

Walk 2:

Enter maximum step

10

Enter boundary

8

Enter x

2

Enter y

1

Walk1: Steps: 0; Position: (0,0)

Walk2: Steps: 0; Position: (2,1)

Walk 1: Steps: 1; Position: (-1,0)

Walk 2: Steps: 1; Position: (2,2)

Walk 1: Steps: 2; Position: (-1,1)

Walk 2: Steps: 2; Position: (3,2)

Walk 1: Steps: 3; Position: (-1,2)

Walk 2: Steps: 3; Position: (3,3)

Walk 1: Steps: 4; Position: (-1,3)

Walk 2: Steps: 4; Position: (2,3)

Walk 1: Steps: 5; Position: (-1,4)

Walk 2: Steps: 5; Position: (2,4)

Walk3 before walk: Steps: 0; Position: (0,0)

Walk3 after walk: Steps: 23; Position: (6,11)

*Add more exercises as needed*

Exercise 4 -- need to submit source code and I/O  
 -- check if completely done \_\_x\_\_ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

**package** pa7;

/\* Java Class: CSCI 145

Author: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**import** java.util.Scanner;

**import** java.text.NumberFormat;

**public** **class** FractionArithmetic {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**final** String add = "+";

**final** String subtract = "-";

**final** String multiple = "\*";

**final** String divide = "/";

**final** String SENTINEL\_VALUE = "%";

**int** questionsAttempted = 0;

**int** totalCorrectAnswer = 0;

**int** num1;

**int** num2;

**int** denom1;

**int** denom2;

String catcher;

String operation;

**int** answerNum;

**int** answerDenom;

RationalNumber input1;

RationalNumber input2;

RationalNumber inputAnswer;

RationalNumber actualAnswer = **new** RationalNumber(0, 1);

Scanner scan = **new** Scanner(System.***in***);

NumberFormat percent = NumberFormat.*getPercentInstance*();

percent.setMaximumFractionDigits(1);

System.***out***.println("Ivan Leung's Rational Tutorial Program\n");

System.***out***.println("Please follow instruction carefully.\nEnter your operation like 1 / 2 + 1 / 4.");

System.***out***.println("You must enter a valid operation.\nEnter operator q to stop the program (0 / 1 % 0 / 1).");

num1 = num2 = denom1 = denom2 = 1;

**do** {

System.***out***.print("\nPlease enter your operation --> ");

num1 = scan.nextInt();

catcher = scan.next();

denom1 = scan.nextInt();

operation = scan.next();

num2 = scan.nextInt();

catcher = scan.next();

denom2 = scan.nextInt();

input1 = **new** RationalNumber(num1, denom1);

input2 = **new** RationalNumber(num2, denom2);

**if** (!operation.trim().equals(SENTINEL\_VALUE)) {

System.***out***.print("Please enter your result --> ");

answerNum = scan.nextInt();

catcher = scan.next();

answerDenom = scan.nextInt();

inputAnswer = **new** RationalNumber(answerNum, answerDenom);

++questionsAttempted;

**switch** (operation.trim()) {

**case** add:

actualAnswer = input1.add(input2);

**break**;

**case** subtract:

actualAnswer = input1.subtract(input2);

**break**;

**case** multiple:

actualAnswer = input1.multiply(input2);

**break**;

**case** divide:

actualAnswer = input1.divide(input2);

**break**;

}

**if** (actualAnswer.compareTo(inputAnswer) == 0) {

System.***out***.println("Great job! It is correct.");

System.***out***.println(input1 + " " + operation + " " + input2 + " = " + inputAnswer);

++totalCorrectAnswer;

}

**else** {

System.***out***.println("It is incorrect:");

System.***out***.println(input1 + " " + operation + " " + input2 + " != " + inputAnswer);

System.***out***.println("The correct answer:");

System.***out***.println(input1 + " " + operation + " " + input2 + " = " + actualAnswer);

}

}

} **while** (!operation.equals(SENTINEL\_VALUE));

System.***out***.println("You have chosen to exit the program");

System.***out***.println("You answered " + totalCorrectAnswer + " out of " + questionsAttempted

+ (questionsAttempted == 0 ? " problem" : " problems")

+ " correctly (" + percent.format( (**double**) totalCorrectAnswer / questionsAttempted) + ").");

scan.close();

}

}

Input/output below:

Ivan Leung's Rational Tutorial Program

Please follow instruction carefully.

Enter your operation like 1 / 2 + 1 / 4.

You must enter a valid operation.

Enter operator q to stop the program (0 / 1 % 0 / 1).

Please enter your operation --> 1 / 2 + 1 / 4

Please enter your result --> 3 / 4

Great job! It is correct.

1/2 + 1/4 = 3/4

Please enter your operation --> 1 / 2 - 1 / 4

Please enter your result --> 1 / 4

Great job! It is correct.

1/2 - 1/4 = 1/4

Please enter your operation --> 1 / 2 \* 1 / 4

Please enter your result --> 1 / 8

Great job! It is correct.

1/2 \* 1/4 = 1/8

Please enter your operation --> 1 / 2 / 1 / 4

Please enter your result --> 2 / 1

Great job! It is correct.

1/2 / 1/4 = 2

Please enter your operation --> -1 / 4 + 1 / 20

Please enter your result --> -4 / 20

Great job! It is correct.

-1/4 + 1/20 = -1/5

Please enter your operation --> 5 / 3 - -1 / 3

Please enter your result --> 2 / 1

Great job! It is correct.

5/3 - -1/3 = 2

Please enter your operation --> 1 / 2 \* 0 / 1

Please enter your result --> 0 / 1

Great job! It is correct.

1/2 \* 0 = 0

Please enter your operation --> 1 / 90 + 2 / 55

Please enter your result --> 47 / 990

Great job! It is correct.

1/90 + 2/55 = 47/990

Please enter your operation --> 0 / 1 % 0 / 1

You have chosen to exit the program

You answered 8 out of 8 problems correctly (100%).

Answer for Question 1

If the method is not static, we would not be able to call the method without instantiate a ChangingPeople object.

Answer for Question 2

All primitive types are always passed by value, which is no arguments here. However, passing object in as parameter is not as straightforward as primitive types. When an object is passed into a method as parameter, the value of an object is passed in which is the memory address of said object. On the one hand, some people may argue that this is an example of pass by value, because the memory address of an object is copied and then passed into a method. On the other hand, passing in the address of an object is actually pass by reference, since a reference (the memory address) of an object is passed into a method.

Extra Credit – provide if applicable

Pseudocode below if applicable:

Source code below:

**package** pa7;

/\* Java Class: CSCI 145

Author: Ivan Leung

Class: Mon/Wed

Date: Apr 11 2023

Description:

I certify that the code below is my own work.

Exception(s): N/A

\*/

**public** **class** RandomWalk {

**private** **int** x;

**private** **int** y;

**private** **int** maxSteps;

**private** **int** currentSteps;

**public** **int** boundary;

**public** RandomWalk(**int** max, **int** edge) {

maxSteps = max;

boundary = edge;

currentSteps = 0;

x = 0;

y = 0;

}

**public** RandomWalk(**int** max, **int** edge, **int** startX, **int** startY) {

maxSteps = max;

boundary = edge;

x = startX;

y = startY;

}

**public** **void** takeStep() {

**int** randomDirection = (**int**) (Math.*random*() \* 3);

**switch** (randomDirection) {

**case** 0:

++x;

++currentSteps;

**break**;

**case** 1:

--x;

++currentSteps;

**break**;

**case** 2:

++y;

++currentSteps;

**break**;

**case** 3:

--y;

++currentSteps;

**break**;

}

}

**public** String toString() {

**return** "Steps: " + currentSteps + "; Position: (" + x + "," + y + ")";

}

**public** **boolean** moreSteps() {

**return** currentSteps < maxSteps;

}

**public** **boolean** inBounds() {

**return** (**int**) Math.*abs*(x) <= boundary && (**int**) Math.*abs*(y) <= boundary;

}

**public** **void** walk() {

**while**(moreSteps() && inBounds()) {

takeStep();

}

}

}

Input/output below:  
  
Enter the max steps: 200

Enter the max boundary: 10

Out of 10 drunks, 10 drunks fell off the platform.

Enter the max steps: 500

Enter the max boundary: 10

Out of 10 drunks, 10 drunks fell off the platform.

Enter the max steps: 200

Enter the max boundary: 50

Out of 10 drunks, 10 drunks fell off the platform.