## CSCI 145 PA \_\_12\_\_ Submission

## Due Date:\_\_\_May 24 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** pa12;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: May 17 2023

Description:

`

I certify that the code below is my own work.

Exception(s): N/A

\*/

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

//ParseInts.java

//

//Reads a line of text and prints the integers in the line.

//

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

**import** java.util.Scanner;

**public** **class** ParseInts {

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

**int** val, sum = 0;

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

// String line;

System.***out***.println("Enter a line of text");

Scanner scanLine = **new** Scanner(scan.nextLine());

**while** (scanLine.hasNext()) {

**try** {

val = Integer.*parseInt*(scanLine.next());

sum += val;

}

**catch** (NumberFormatException e) {

}

}

scan.close();

scanLine.close();

System.***out***.println("The sum of the integers on this line is " + sum);

}

}

Input/output below:

Enter a line of text

We have 5 dogs, 2 cats, and 10 fishes.

The sum of the integers on this line is 17

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** pa12;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: May 17 2023

Description:

`

I certify that the code below is my own work.

Exception(s): N/A

\*/

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

//Warning.java

//

//Reads student data from a text file and writes data to another text file.

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

**import** java.util.Scanner;

**import** java.io.\*;

**import** java.text.DecimalFormat;

**public** **class** Warning {

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

// Reads student data (name, semester hours, quality points) from a

// text file, computes the GPA, then writes data to another file

// if the student is placed on academic warning.

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

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

**int** creditHrs;

// number of semester hours earned

**double** qualityPts;

// number of quality points earned

**double** gpa;

// grade point (quality point) average

DecimalFormat decimal = **new** DecimalFormat("#.00");

String line, name, inputName = "C:\\Users\\ivanl\\OneDrive\\Desktop\\dev\\broken\_code\\csci\_145\\hw\\pa\\pa12\\students.dat";

String outputName = "C:\\Users\\ivanl\\OneDrive\\Desktop\\dev\\broken\_code\\csci\_145\\hw\\pa\\pa12\\warning.dat";

**try** {

// Set up scanner to input file

Scanner fileScan = **new** Scanner(**new** File(inputName));

// Set up the output file stream

PrintWriter outFile = **new** PrintWriter(outputName);

// Print a header to the output file

outFile.println();

outFile.println("Students on Academic Warning");

outFile.println();

// Process the input file, one token at a time

**while** (fileScan.hasNext()) {

// Get the credit hours and quality points and

name = fileScan.next();

creditHrs = fileScan.nextInt();

qualityPts = fileScan.nextDouble();

// determine if the student is on warning. If so,

gpa = qualityPts / creditHrs;

// write the student data to the output file.

**if** ((creditHrs < 30 && gpa < 1.5) || (creditHrs < 60 && gpa < 1.75) || (gpa < 2.0)) {

outFile.println(name + " " + creditHrs + " " + decimal.format(gpa));

}

}

// Close output file

fileScan.close();

outFile.close();

} **catch** (FileNotFoundException exception) {

System.***out***.println("The file " + inputName + " was not found.");

} **catch** (IOException exception) {

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

} **catch** (NumberFormatException e) {

System.***out***.println("Format error in input file: " + e);

}

}

}

Input/output below:

students.dat

Students on Academic Warning

Jones 21 1.35

Walker 96 1.90

Street 33 1.74

Davis 110 1.80

Summers 52 1.60

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** pa12;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: May 17 2023

Description:

`

I certify that the code below is my own work.

Exception(s): N/A

\*/

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

//BaseConversion.java

//

//Recursively converts an integer from base 10 to another base

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

**import** java.util.Scanner;

**public** **class** BaseConversion {

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

**int** base10Num;

**int** base;

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

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

System.***out***.println("Base Conversion Program");

System.***out***.print("Enter an integer: ");

base10Num = scan.nextInt();

System.***out***.print("Enter the base: ");

base = scan.nextInt();

// Call convert and print the answer

System.***out***.println(*convert*(base10Num, base));

}

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

// Converts a base 10 number to another base.

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

**public** **static** String convert(**int** num, **int** b) {

**int** quotient = num / b; // the quotient when num is divided by base b

**int** remainder = num % b; // the remainder when num is divided by base b

**if** (quotient == 0) {

**return** (b > 9 && remainder > 9) ? "" + (**char**) (remainder + 55) : "" + remainder;

}

**return** (b > 9 && remainder > 9)

? *convert*(quotient, b) + (**char**) (remainder + 55) : *convert*(quotient, b) + remainder;

}

}

Input/output below:

Base Conversion Program

Enter an integer: 65500

Enter the base: 16

FFDC

*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** pa12;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: May 17 2023

Description:

`

I certify that the code below is my own work.

Exception(s): N/A

\*/

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

//Backwards.java

//

//Uses a recursive method to print a string backwards.

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

**import** java.util.Scanner;

**public** **class** Backwards {

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

// Reads a string from the user and prints it backwards.

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

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

String msg;

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

System.***out***.print("Enter a string: ");

msg = scan.nextLine();

System.***out***.print("\nThe string backwards: ");

*printBackwards*(msg);

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

}

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

// Takes a string and recursively prints it backwards.

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

**public** **static** **void** printBackwards(String s) {

// Fill in code

**if** (s.length() < 1)

**return**;

System.***out***.print(s.charAt(s.length() - 1));

*printBackwards*(s.substring(0, s.length() - 1));

}

}

Input/output below:

Enter a string: Hello Java!

The string backwards: !avaJ olleH

Answer for Question 1

By handling exceptions, we ensure that the program can continue to run without interruption. Most of the time, we do not want the program to be terminated. Another reason is that when exceptions are being caught, the program can generate error report which can be used later for debugging.

Answer for Question 2

In some cases, recursive solutions can be easily implemented when compared to iterative solutions. Also, recursive solutions usually have better readability than iterative solutions. However, iterative solutions almost always perform better than recursive solutions such as less run time and use less memory.

Extra Credit – provide if applicable

Pseudocode below if applicable:

Source code below:

**package** pa12;

/\* Java Class: CSCI 145

Modified by: Ivan Leung

Class: Mon/Wed

Date: May 17 2023

Description:

`

I certify that the code below is my own work.

Exception(s): N/A

\*/

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.util.Scanner;

**public** **class** GolfScores {

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

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

**int** min, currentPar, totalPar = 0;

**int** playerScore1 = 0;

**int** playerScore2 = 0;

**int** playerScore3 = 0;

String fileName = "C:\\Users\\ivanl\\OneDrive\\Desktop\\dev\\broken\_code\\csci\_145\\hw\\pa\\pa12\\golf3players.txt";

String par = "";

String player1 = "";

String player2 = "";

String player3 = "";

Scanner fileScan;

**try** {

fileScan = **new** Scanner(**new** File(fileName));

par = fileScan.next();

player1 = fileScan.next();

player2 = fileScan.next();

player3 = fileScan.next();

**while** (fileScan.hasNext()) {

currentPar = fileScan.nextInt();

totalPar += currentPar;

playerScore1 += fileScan.nextInt() - currentPar;

playerScore2 += fileScan.nextInt() - currentPar;

playerScore3 += fileScan.nextInt() - currentPar;

}

fileScan.close();

} **catch** (FileNotFoundException exception) {

System.***out***.println("The file " + fileName + " was not found.");

} **catch** (IOException exception) {

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

} **catch** (NumberFormatException e) {

System.***out***.println("Format error in input file: " + e);

}

// Find the winner

min = playerScore1;

**if** (playerScore2 < min)

min = playerScore2;

**if** (playerScore3 < min)

min = playerScore3;

System.***out***.printf("%-10s%d%n", par + ":" , totalPar);

System.***out***.printf("%-10s%s%s%n", player1 + ":", (playerScore1 > -1 ? "+" : "") + playerScore1, (playerScore1 == min ? " (winner)" : ""));

System.***out***.printf("%-10s%s%s%n", player2 + ":", (playerScore2 > -1 ? "+" : "") + playerScore2, (playerScore2 == min ? " (winner)" : ""));

System.***out***.printf("%-10s%s%s%n", player3 + ":", (playerScore3 > -1 ? "+" : "") + playerScore3, (playerScore3 == min ? " (winner)" : ""));

}

}

Input/output below:

Par: 63

Lee: -12 (winner)

Smith: +3

Kim: +2