Christopher Bussen

CPS 150 02 – Algorithms and Programming 1

Assignment 5

11/23/20

**Problem 1 – R5.5**

1. 39.0
2. 400.0
3. 92.0
4. 62.0
5. 0.0

**Problem 2 – R5.10**

The first b and the first i are restricted to the main method. The second i and the first n are restricted to the f method. The variable a and the second b are restricted to the g method. Finally, the second i and the second n are restricted to the for loop in the g method.

This program will print 26.

**Problem 3 – R5.14**

The falseSwap method does not successfully swap the contents of x and y when called in the main program because it does not return any value. As a result, when called, it will execute and swap the values, but it won’t return these new swapped values to the main method, and the original x and y from the main method will be printed rather than the swapped values. This could be fixed by moving the print statement from the main method to the falseSwap method. This way, the print statement will execute when the two integers have been swapped, but before the method ends and returns nothing to the main method.

**Problem 4 – E5.1**

1. Smallest

RUNNING SCREENSHOT

Text

Description automatically generated  
CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

SmallestMethod: number number number ; number

program takes in 3 integers and determines/prints the smallest of the 3

ex1: user inputs 1, 2, 3 - program outputs 1

ex2: user inputs 88, 61, -900 - program outputs -900

ex3: user inputs 18, 19, 18- program outputs 18

ex4: user inputs x, y, z - program outputs error

ex5: user inputs 2.4, 3.1, 6 - program outputs error

\*/

import java.util.Scanner;

public class SmallestMethod {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter 3 integers and declare int variables for each of them

System.out.print("Enter 3 integers: ");

int num1 = input.nextInt();

int num2 = input.nextInt();

int num3 = input.nextInt();

//call the minimum method in a print statement using the 3 user numbers as inputs

System.out.println("The minimum is " + findMinimum(num1, num2, num3));

}

//create separate method to determine the smallest of three numbers - takes in 3 ints and returns an int

public static int findMinimum(int a, int b, int c){

//declare an int variable for the minimum

int minimum = 0;

//use if statements to determine the minimum

if(a <= b && a <= c){

minimum = a;

}

else if(b <= a && b <= c){

minimum = b;

}

else{

minimum = c;

}

//return the minimum

return minimum;

}

}

1. Largest

RUNNING SCREENSHOT

Text

Description automatically generated

CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

LargestMethod: number number number ; number

program takes in 3 integers and determines/prints the largest of the 3

ex1: user inputs 1, 2, 3 - program outputs 3

ex2: user inputs 88, 61, -900 - program outputs 88

ex3: user inputs 18, 19, 19- program outputs 19

ex4: user inputs x, y, z - program outputs error

ex5: user inputs 2.4, 3.1, 6 - program outputs error

\*/

import java.util.Scanner;

public class LargestMethod {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter 3 integers and declare int variables for each of them

System.out.print("Enter 3 integers: ");

int num1 = input.nextInt();

int num2 = input.nextInt();

int num3 = input.nextInt();

//call the maximum method in a print statement using the 3 user numbers as inputs

System.out.println("The maximum is " + findMaximum(num1, num2, num3));

}

//create separate method to determine the largest of three numbers - takes in 3 ints and returns an int

public static int findMaximum(int a, int b, int c){

//declare an int variable for the maximum

int maximum = 0;

//use if statements to determine the maximum

if(a >= b && a >= c){

maximum = a;

}

else if(b >= a && b >= c){

maximum = b;

}

else{

maximum = c;

}

//return the maximum

return maximum;

}

}

**Problem 5 – E5.7**

RUNNING SCREENSHOT

Text

Description automatically generated

CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

WordCount: string ; number

program takes in a string from the user and determines and prints out the

number of words in the string

ex1: user inputs Mary had a little lamb - program outputs 5

ex2: user inputs I love to write - program outputs 4

ex3: user inputs x y z baby tissue towel - program outputs 6

ex4: user inputs charger - program outputs 1

\*/

import java.util.Scanner;

public class WordCount {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter a string and declare a string variable for the value

System.out.print("Please enter your string: ");

String message = input.nextLine();

//call the word counting method with the user string as an input and print the number of words

System.out.println("There are " + countWords(message) + " words in your string");

}

//create separate method to determine number of words in string - takes in a string and returns an int

public static int countWords(String a){

//declare an int variable for the word counter - start at 1

int wordCount = 1;

//if first character is a space, subtract one from the count

if(a.charAt(0) == ' '){

wordCount--;

}

//if final character is a space, subtract one from count

if(a.charAt(a.length()-1) == ' '){

wordCount--;

}

//declare a boolean variable for whether or not the last character was a space

boolean recentSpace = false;

//use for loop to determine number of words - loop through the string and find spaces

for(int i = 0; i < a.length(); i++){

if(a.charAt(i) == ' ' && !recentSpace){

//if character is a space, add 1 to word count

wordCount++;

//change boolean variable for last character being a space to true

recentSpace = true;

}

//when character is not a space, change boolean variable for last character being a space to false

if(a.charAt(i) != ' '){

recentSpace = false;

}

}

//return the word count

return wordCount;

}

}

**Problem 6 – E5.10**

RUNNING SCREENSHOT

Text

Description automatically generated

CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

RecursiveReverseOfString: string ; string

program takes in a string and determines/prints the reverse

of the string

ex1: user inputs flow - program outputs wolf

ex2: user inputs water - program outputs retaw

ex3: user inputs 1234 - program outputs 4321

ex4: user inputs book bag - program outputs gab koob

\*/

import java.util.Scanner;

public class RecursiveReverseOfString {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter a string and declare a string variable for the value

System.out.print("Enter your string: ");

String str = input.nextLine();

//call method to reverse string using user input as parameter and print the reversed string

System.out.println("The reversed string is " + reverse(str));

}

//create a separate method to reverse the string

public static String reverse(String a){

if(a.length() == 0){

return a;

}

//call recursive method to return string without first letter then add first letter onto end

return reverse(a.substring(1)) + a.charAt(0);

}

}

**Problem 7 – P5.15**

RUNNING SCREENSHOT

Text

Description automatically generated

CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

AddingFractions: number number number number ; number number

program takes in a numerator and denominator for two fractions and calculates

and prints out the simplified sum of the two fractions

ex1: user inputs 3, 4, 5, 6 - program outputs 19/12

ex2: user inputs 1, 2, 1, 2 - program outputs 1/1

ex3: user inputs 7, 99, -1, 4 - program outputs -71/396

ex4: user inputs x, y, z, f - program outputs error

ex5: user inputs 10, 31, 9, 0 - program outputs error

\*/

import java.util.Scanner;

public class AddingFractions {

public static void main(String [] args){

//import the scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter the numerator and denominator for the first fraction and declare int variables for them

System.out.print("Enter the numerator of the first fraction: ");

int firstNumerator = input.nextInt();

System.out.print("Enter the denominator of the first fraction: ");

int firstDenominator = input.nextInt();

//prompt the user to enter the numerator and denominator for the second fraction and declare int variables for them

System.out.print("Enter the numerator of the second fraction: ");

int secondNumerator = input.nextInt();

System.out.print("Enter the denominator of the second fraction: ");

int secondDenominator = input.nextInt();

//call method that adds fractions using two numerators and denominators as inputs and print results

System.out.print(firstNumerator + "/" + firstDenominator + " + " + secondNumerator + "/" + secondDenominator + " = ");

System.out.println(addFractions(firstNumerator, firstDenominator, secondNumerator, secondDenominator));

}

//create a separate method that finds the greatest common divisor of two numbers - takes 2 ints as input and returns an int

public static int greatestCommonDivisor(int a, int b){

//return a when it is 0

if(a == 0){

return b;

}

//call recursive method

return greatestCommonDivisor(b%a, a);

}

//create a separate method to simplify the added fractions

public static String simplifyFractions(int x, int y){

//find the greatest common divisor of the two ints

int commonDenom = greatestCommonDivisor(x, y);

//divide fractions by their common denominator

x = x / commonDenom;

y = y / commonDenom;

//print the simplified fraction

return (x + "/" + y);

}

//create a separate method to add the fractions

public static String addFractions(int numerator1, int denominator1, int numerator2, int denominator2){

//find the denominator by calling the greatest common divisor method for the two denominators

int finalDenom = greatestCommonDivisor(denominator1, denominator2);

//find least common multiple of two denominators

finalDenom = (denominator1 \* denominator2) / finalDenom;

//add the numerators of the two fractions

int finalNumerator = numerator1 \* (finalDenom / denominator1) + numerator2 \* (finalDenom / denominator2);

//call method that simplifies the fraction using the final numerator and final denominator

return simplifyFractions(finalNumerator, finalDenom);

}

}

**Problem 8 – Business P5.27**

RUNNING SCREENSHOT

Text

Description automatically generated

CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

ValidPassword: string string ; string

program takes in a password string from a user, a confirmation of the same password,

and determines if the password is valid based on the following rules: at least 8 characters

long, has at least one uppercase and lowercase letter, and has at least one digit

ex1: user inputs Password1, Password 1 - program outputs Password is valid

ex2: user inputs waterBottle2, waterBottle2 - program outputs Password is valid

ex3: user inputs elephantS1, elephantG1 - program outputs Password not valid - passwords must match, be 8+ characters, include an uppercase and lowercase letter, and include a digit

ex4: user inputs abcdefg123, abcdefg123 - program outputs Password not valid - passwords must match, be 8+ characters, include an uppercase and lowercase letter, and include a digit

ex5: user inputs iPhoneLover, iPhoneLover - program outputs Password not valid - passwords must match, be 8+ characters, include an uppercase and lowercase letter, and include a digit

ex6: user inputs Pass12, Pass12 - program outputs Password not valid - passwords must match, be 8+ characters, include an uppercase and lowercase letter, and include a digit

\*/

import java.util.Scanner;

public class ValidPassword {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter a password and create a string variable for it

System.out.print("Enter your password: ");

String password = input.nextLine();

//prompt the user to confirm the password and create a new string variable for this

System.out.print("Confirm your password: ");

String confirmation = input.nextLine();

//call method that checks password and tell user whether or not password is valid - use password and confirmation as inputs

if(checkPassword(password, confirmation)){

System.out.println("Password is valid");

}

else{

System.out.println("Password not valid - passwords must match, be 8+ characters, include an uppercase and lowercase letter, and include a digit");

}

}

//create separate method that checks if password is valid - takes two strings as input and returns boolean

public static boolean checkPassword(String a, String b){

//declare boolean variable for whether or not the passwords are valid

boolean valid = true;

//declare boolean variables for each requirement

boolean confirmationEqual = true;

boolean length = true;

boolean upperCase = true;

boolean lowerCase = true;

boolean digit = true;

//check if password and confirmation (a and b) are the same - if not, change valid to false

if(!(a.equals(b))){

confirmationEqual = false;

}

//check if password (a) is at least 8 characters long - if not, change valid to false

if(!(a.length() >= 8)){

length = false;

}

//check if password has at least one uppercase letter - if not, change valid to false

for(int i = 0; i < a.length(); i++){

char x = a.charAt(i);

if(Character.isUpperCase(x)){

upperCase = true;

break;

}

else{

upperCase = false;

}

}

//check if password has at least one lowercase letter - if not, change valid to false

for(int j = 0; j < a.length(); j++){

char y = a.charAt(j);

if(Character.isLowerCase(y)){

lowerCase = true;

break;

}

else{

lowerCase = false;

}

}

//check if password has at least one digit - if not, change valid to false

for(int k = 0; k < a.length(); k++){

char z = a.charAt(k);

if(z == '0' || z == '1' || z == '2' || z == '3' || z == '4' || z == '5' || z == '6' || z == '7' || z == '8' || z == '9'){

digit = true;

break;

}

else{

digit = false;

}

}

if(!confirmationEqual || !length || !upperCase || !lowerCase || !digit){

valid = false;

}

return valid;

}

}

**Problem 9 – Business P5.33**

RUNNING SCREENSHOT

Text

Description automatically generated

CODE

/\*

Christopher Bussen

CPS 150 02

Assignment 5

HorsepowerToOvercomeDrag: number ; number

program takes in the velocity of the car from the user and calculates then prints

the power in watts and horsepower needed to overcome the resulting drag force

Drag force = (1/2)\*(air density)\*(velocity)^2\*(area of car)\*(drag coefficient)

Power in watts = (Drag force)\*(velocity)

Horsepower = (Power) / 746

ex1: user inputs 20 - program outputs 2460.0, 3.297

ex2: user inputs 89 - program outputs 216777.9675, 290.587

ex3: user inputs 5 - program outputs 38.4375, 0.051

ex4: user inputs orange - program outputs error

ex5: user inputs 0 - program outputs 0.0, 0.0

\*/

import java.util.Scanner;

public class HorsepowerToOvercomeDrag {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter the velocity of the car and declare a double variable for it

System.out.print("Enter the velocity of the car (m/s): ");

double velocity = input.nextDouble();

//declare a double variable for drag force by calling drag force method using user input velo

double dragForce = computeDragForce(velocity);

//declare a double variable for power and calculate power in watts needed

double power = dragForce \* velocity;

//declare a double variable for the horsepower and calculate the horsepower needed

double horsepower = power / 746;

//print both power and horsepower needed

System.out.println(power + " watts or " + horsepower + " hp are needed to overcome the drag force");

}

//create a separate method to calculate the drag force on the car - takes in a double for velocity and returns a double

public static double computeDragForce(double carVelo){

//declare a double variable for the value of air density

double airDensity = 1.23;

//declare a double variable for the value of the car's area

double carArea = 2.5;

//declare a double variable for the drag coefficient

double dragCoefficient = 0.2;

//declare a double variable for drag force and return this variable

double drag = (1.0/2) \* (airDensity) \* (Math.pow(carVelo, 2)) \* (carArea) \* (dragCoefficient);

return drag;

}

}