**Q1.**

package week\_3;

import java.util.Scanner;

/\*\*

\* Write a program to calculate the MPG for a car journey.

\* (MPG = Miles per gallon, calculate by dividing number of miles, by number of gallons of gas used).

\*

\* Write a method called mpg that has two parameters, in the following order:

\* - a double to represent the number of miles driven, and

\* - a double to represent the number of gallons of gas used on a car journey

\*

\* And returns a double value, representing the MPG for the journey.

\*

\* Your main method should:

\* - Ask the user for the miles and gas used

\* - Call your new mpg method (that you'll write below) and save the returned miles-per-gallon value

\* - Use the returned value to display the MPG

\*/

public class Question\_1\_MPG {

public static void main(String[] args) {

// Create a Scanner object to read user input from the console

Scanner scanner = new Scanner(System.in);

// Prompt the user for the number of miles driven

System.out.print("Enter the number of miles driven: ");

double miles = scanner.nextDouble();

// Prompt the user for the number of gallons of gas used

System.out.print("Enter the number of gallons of gas used: ");

double gallons = scanner.nextDouble();

// Call the mpg method to calculate the miles-per-gallon

double mpg = mpg(miles, gallons);

// Print the miles-per-gallon to the console, with two decimal places

System.out.printf("The miles-per-gallon for this journey is: %.2f", mpg);

// Close the scanner object

scanner.close();

}

/\*\*

\* Calculate the miles-per-gallon for a car journey.

\*

\* @param miles - The number of miles driven

\* @param gallons - The number of gallons of gas used

\* @return The miles-per-gallon for the journey

\*/

public static double mpg(double miles, double gallons) {

// Calculate the miles-per-gallon by dividing miles by gallons

double mpg = miles / gallons;

// Return the miles-per-gallon value

return mpg;

}

}

**Q2.**

package week\_3;

import static input.InputUtils.doubleInput;

/\*\*

\* You have a client who likes to wear a hat when the temperature is at or below 40F.

\* Write a method called doYouNeedAHat which takes one parameter, the current temperature in Fahrenheit, as a double.

\*

\* This method will return true if the temperature is at or below 40F.

\* This method will return false if the temperature is above 40F.

\*/

public class Question\_2\_Wear\_A\_Hat {

public static void main(String[] args) {

// Get today's temperature in Fahrenheit from the user.

double todayTemperature = doubleInput("Enter today's temperature, in fahrenheit.");

// Call the doYouNeedAHat method with todayTemperature as the argument,

// and store the result in the needsHat variable.

boolean needsHat = doYouNeedAHat(todayTemperature);

// Use the value returned from doYouNeedAHat to print a message

// to the user, telling them whether or not they need a hat.

if (needsHat) {

System.out.println("It's " + todayTemperature + " degrees Fahrenheit today. You need a hat!");

} else {

System.out.println("It's " + todayTemperature + " degrees Fahrenheit today. You don't need a hat.");

}

}

/\*\*

\* This method determines whether or not the user needs a hat based on the temperature.

\*

\* @param temperature the current temperature in Fahrenheit

\* @return true if the temperature is at or below 40F, false otherwise

\*/

public static boolean doYouNeedAHat(double temperature) {

// If the temperature is at or below 40F, return true.

if (temperature <= 40) {

return true;

} else {

// If the temperature is above 40F, return false.

return false;

}

}

}

**Q3.**

package week\_3;

import static input.InputUtils.\*;

/\*\*

\*

\* The fare to ride the bus during regular (not rush) hours is $2.00.

\* The fare to ride the bus during rush hours is $2.50.

\*

\* Create a program to calculate the total spent on bus rides,

\* from the number of regular and number of rush hour trips taken.

\*/

public class Question\_3\_Bus\_Fares {

public static void main(String[] args) {

int regularFareBusRides = positiveIntInput("How many times did you ride the bus during regular hours?");

int rushFareBusRides = positiveIntInput("How many times did you ride the bus during rush hours?");

// Call fareTotal method with the arguments regularFareBusRides and rushFareBusRides

// to calculate the total amount spent on bus rides.

double totalFare = fareTotal(regularFareBusRides, rushFareBusRides);

// Print the total amount spent on all bus rides.

System.out.println("Total spent on all bus rides: $" + totalFare);

}

/\*\*

\* Calculates the total amount spent of bus rides.

\* A regular bus fare is $2, a rush hour bus fare is $2.50.

\*

\* @param regularRides the number of regular bus rides made

\* @param rushRides the number of rush hour bus rides made

\* @return the total amount spent on bus rides

\*/

public static double fareTotal(int regularRides, int rushRides) {

double regularFare = 2.0;

double rushFare = 2.5;

double totalRegularFare = regularFare \* regularRides;

double totalRushFare = rushFare \* rushRides;

double totalFare = totalRegularFare + totalRushFare;

return totalFare;

}

}

**Q4.**

package week\_3;

import static input.InputUtils.\*;

/\*\*

\* Write a program that can help decide if a particular programming project

\* is best solved using a Waterfall or Agile methodology.

\*

\* Your program should ask the user:

\*

\* • How many programmers will be on the team [ More than 30 programmers -> Waterfall ]

\* • If there needs to be firm deadlines and a fixed schedule [ Yes - > Waterfall ]

\* • If the programmers have experience in requirements, analysis and testing as well as coding [ Yes - > Agile ]

\* • If there are stringent quality control requirements [ Yes -> Waterfall ]

\* • If early integration is desirable [ Yes -> Agile ]

\* • If the customer will be requiring working models early in the process [ Yes -> Agile ]

\*

\* There's a `yesNoInput` method in the InputUtils library that returns boolean values from yes/no user input.

\* (If the user types 'n' or 'no', the method returns false. If the user types 'y' or 'yes' the method returns true.)

\*

\* Write a method called agileOrWaterfall,

\* which takes this data as integer and boolean parameters.

\* The parameters should be provided in the order given above.

\* agileOrWaterfall will return a String, a suggestion on whether Agile, or Waterfall, or either, may be is best.

\*

\* To decide, check how many factors are in favor of Agile.

\* If there are 4 or more factors in favor of Agile, then return `AGILE`.

\* If there are 4 or more factors in favor of Waterfall, return `WATERFALL`.

\* If there are an equal number of factors in favor of Agile and Waterfall, returns `EITHER`.

\*

\* Notice that there are three global constants AGILE, WATERFALL and EITHER.

\* Your agileOrWaterfall method should return one of these Strings.

\*

\* Use your agileOrWaterfall method in your program to suggest which methodology to use.

\*

\* Your main method should do the task of asking questions and printing the result.

\* Your agileOrWaterfall method should be given the relevant data, and do the processing,

\* deciding, and returning the result.

\*/

public class Question\_4\_Agile\_Or\_Waterfall {

// You will use these constants in your program to represent the decision on methodology

public static final String AGILE = "Agile";

public static final String WATERFALL = "Waterfall";

public static final String EITHER = "Either";

public static void main(String[] args) {

// Ask the user the 6 questions

int numProgrammers = positiveIntInput("How many programmers will be on the team?");

boolean hasFirmDeadlines = yesNoInput("Will there be firm deadlines and a fixed schedule?");

boolean hasExperience = yesNoInput("Do the programmers have experience in requirements, analysis, and testing as well as coding?");

boolean hasQualityControl = yesNoInput("Are there stringent quality control requirements?");

boolean hasEarlyIntegration = yesNoInput("Is early integration desirable?");

boolean requiresWorkingModels = yesNoInput("Will the customer require working models early in the process?");

// Call your agileOrWaterfall method, passing it the data obtained from the user as arguments

String methodology = agileOrWaterfall(numProgrammers, hasFirmDeadlines, hasExperience, hasQualityControl, hasEarlyIntegration, requiresWorkingModels);

// Use the suggestion agileOrWaterfall returns to print a message for the user.

System.out.println("Based on your answers, " + methodology + " methodology is recommended.");

}

AgileWaterfallGUI() {

setTitle("Agile or Waterfall?");

setContentPane(mainPanel);

pack();

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

setVisible(true);

//TODO any GUI configuration needed

//TODO add event handler to read the data entered, and selections made,

//TODO recommend a methodology, display in JLabel.

// Use the recommendationTemplate to display a String like "Android App should use Agile"

recommendMethodology.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Ask the user the 6 questions

int numProgrammers = peopleOnTeam.getValue();

boolean hasFirmDeadlines = firmDeadlines.isSelected();

boolean hasExperience = experienceAllPhases.isSelected();

boolean hasQualityControl = qualityControl.isSelected();

boolean hasEarlyIntegration = earlyIntegration.isSelected();

boolean requiresWorkingModels = earlyWorkingModels.isSelected();

// Call your agileOrWaterfall method, passing it the data obtained from the user as arguments

String methodology = agileOrWaterfall(numProgrammers, hasFirmDeadlines, hasExperience, hasQualityControl, hasEarlyIntegration, requiresWorkingModels);

// Use the suggestion agileOrWaterfall returns to print a message for the user.

System.out.println("Based on your answers, " + methodology + " methodology is recommended.");

}

public static String agileOrWaterfall(int numProgrammers, boolean firmDeadlines, boolean reqAnalysisTesting,

boolean stringentQC, boolean earlyIntegration, boolean customerEarlyModels) {

int agileFactors = 0;

int waterfallFactors = 0;

if (numProgrammers > 30) {

waterfallFactors++;

} else {

agileFactors++;

}

if (firmDeadlines) {

waterfallFactors++;

} else {

agileFactors++;

}

if (reqAnalysisTesting) {

agileFactors++;

} else {

waterfallFactors++;

}

if (stringentQC) {

waterfallFactors++;

} else {

agileFactors++;

}

if (earlyIntegration) {

agileFactors++;

} else {

waterfallFactors++;

}

if (customerEarlyModels) {

agileFactors++;

} else {

waterfallFactors++;

}

if (agileFactors >= 4) {

return AGILE;

} else if (waterfallFactors >= 4) {

return WATERFALL;

} else {

return EITHER;

}

}

});

}

}

**Q5.**

package week\_3;

import java.util.Arrays;

/\*\*

Write a program that can convert arrays to uppercase or lowercase.

You will create a method that can convert an array of Strings to uppercase or lowercase.

As with all lab questions, you should COMMENT YOUR CODE. As well as the tests, the instructor will also read your comments

to verify that you understand and can describe the code you've written.

\*/

public class Question\_5\_Change\_Array\_Case {

public static void main(String[] args) {

// Example arrays to test the convertArrayCase method

String[] test1 = { "lower", "UPPER", "MiXeD" };

String[] test2 = { "a", "b", "c", "d" };

String[] test3 = { "A", "B", "C", "D" };

// Testing convertArrayCase method with the example arrays

convertArrayCase(test1, true);

System.out.println(Arrays.toString(test1)); // expect to print { "LOWER", "UPPER", "MIXED" }

convertArrayCase(test1, false);

System.out.println(Arrays.toString(test1)); // { "lower", "upper", "mixed" }

convertArrayCase(test2, true);

System.out.println(Arrays.toString(test2)); // expect to print { "A", "B", "C", "D" }

convertArrayCase(test2, false);

System.out.println(Arrays.toString(test2)); // expect to print { "a", "b", "c", "d" }

convertArrayCase(test3, true);

System.out.println(Arrays.toString(test3)); // expect to print { "A", "B", "C", "D" }

convertArrayCase(test3, false);

System.out.println(Arrays.toString(test3)); // expect to print { "a", "b", "c", "d" }

}

/\*\*

\* A method that can convert an array of Strings to uppercase or lowercase.

\*

\* @param array The String array to be converted.

\* @param toUpperCase A boolean indicating whether to convert to uppercase or lowercase.

\*/

public static void convertArrayCase(String[] array, boolean toUpperCase) {

// Loop through the array and convert each element to uppercase or lowercase

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

if (toUpperCase) {

array[i] = array[i].toUpperCase();

} else {

array[i] = array[i].toLowerCase();

}

}

}

}

**Q6.**

package week\_3;

import static input.InputUtils.doubleInput;

/\*\*

\* Marathon

\*

\* You have a friend who wants to run a marathon race, which is 26.2 miles.

\* Currently, your friend can run exactly 1 mile. Your friend's plan is to

\* try to run 10% further every week, until they can run the entire

\* marathon distance.

\*

\* So in the

\* first week of training, they will run 1 mile;

\* second week 1.1 miles,

\* third week 1.21 miles...

\*

\* You would like to write a program to print out their training schedule.

\*

\* But, what if your friend thinks this schedule is too slow? Or too fast?

\* Your friend can get to the marathon distance sooner by increasing their distance

\* by 15% every week? Or, they may want to be more leisurely and increase by 3% every week.

\* How does that affect the number of weeks needed?

\*

\* Then, you think, what if your friend decides they want to train for a

\* 10 mile race? Or a 100 mile race?

\*

\* What if your friend starts with the ability to run 5 miles?

\*

\* Write a method that takes 3 parameters:

\* - the current distance your friend can run,

\* - the target distance

\* - the percent increase. This should be expressed as 5 for 5%, 10 for 10%.

\*

\* In your method, use a loop to calculate the distance they need to run every

\* week until they can run the target distance.

\*

\* The method should also print the weekly distance data as a table. You can do this in the loop.

\* Use string formatting to display the weekly distances to 2 decimal places.

\*

\* The method should return the number of weeks it will take, as an integer.

\*/

public class Question\_6\_Marathon {

public static void main(String[] args) {

double currentDistance = doubleInput("How far can you run right now, in miles?");

double targetDistance = doubleInput("What is the target distance, in miles?");

double weeklyPercentIncrease = doubleInput("What percent further do you want to run every week? Example: enter 5 for 5% or 10 for 10% ");

// Call your trainingSchedule method.

int weeks = trainingSchedule(currentDistance, targetDistance, weeklyPercentIncrease);

// Print the total number of weeks.

System.out.println("Total number of weeks: " + weeks);

}

// Write a public static method called trainingSchedule

// It will take 3 double parameters: currentDistance, targetDistance, and percentIncrease

// Ensure that you use three double parameters, in this order.

//

// The method will print a table of weeks and distances.

// The final row should be a distance greater or equal to the target distance.

// the table columns should be in that order (weeks then distance)

//

// Example table for start 1 miles, target 20, increase 10%

//

// Week Distance

// 1 1.0

// 2 1.10

// 3 1.21

// 4 1.33

//

// ..... more rows here .....

//

// 31 17.45

// 32 19.19

// 33 21.11

public static int trainingSchedule(double currentDistance, double targetDistance, double weeklyPercentIncrease) {

int week = 1;

double distance = currentDistance;

System.out.printf("%-8s%s%n", "Week", "Distance");

while (distance < targetDistance) {

System.out.printf("%-8d%.2f%n", week, distance);

distance += distance \* weeklyPercentIncrease / 100.0;

week++;

}

System.out.printf("%-8d%.2f%n", week, distance);

return week;

}

}

//

//

// Note that the last line is the first distance greater than or equal to the target distance.

// Use String formatting to display distances to 2 decimal places. Use String.format() or System.out.printf().

//

// Finally, your method should return the total number of weeks in the training schedule